EXPLOITING QUERY FEATURES IN LANGUAGE MODELING APPROACH FOR INFORMATION RETRIEVAL

by

Munirathnam Srikanth

26 August 2004

A dissertation submitted to the
Faculty of the Graduate School of
The State University of New York at Buffalo
in partial fulfilment of the requirements for the
degree of

Doctor of Philosophy

Department of Computer Science and Engineering

To Jaya

Acknowledgements

I wish to express my deepest gratitude to my advisor, Rohini Srihari. From the beginning, she has treated me as a peer and a friend and given me the freedom and guidance to explore different problems in my research area. She encouraged me to participate in different aspects of research from problem identification, to proposing solutions, to planning experiments and evaluations, to technical writing. Over the years I have learned to truely appreciate good research and, with an emphasis on experimentation and evaluation, recognize and solve real research problems.

I would like to thank the other members of my committee, Jan Chomicki and David Pierce. Jan has provided very useful comments that helped me clarify some of the aspects of my thesis. David has been very helpful since the time of my thesis proposal. His suggestion to explore some NLP aspects in language modeling approaches to information retrieval prompted me to pursue and propose concept language models. I have benefited from a number of discussions we have had on information extraction and question answering. I would like to thank my external reader, ChengXiang Zhai, for his thoughtful comments and suggestions on my dissertation.

Special thanks is due to Miguel Ruiz for encouraging my explorations in cross-lingual retrieval and collaborating with me on a number of projects. I would like to thank Sargur Srihari for providing a stimulating research environment at CEDAR to collaborate and discuss different research problems in pattern recognition and document analysis. Thanks to all my friends and collegues at CEDAR with whom I spent my time as graduate student at University at Buffalo. In particular, I would like to thank Xiaoyun Wu and Ajay

Shekhawat with whom I shared a lot of good times discussing a number of aspects of life and research. Thanks to Zhaohui Zhang, Wei Dai, Sudarshan Lamkhede, Niranjan Balasubramanian, Himanshu Ashiya for stimulating technical discussions. I would like to thank Cymfony Inc. for providing access to their information extraction tool kit (InfoXtract) for some of my retrieval experiments.

Initial seeds of my desire to pursue scientific inquiry were sown by my late grandfather – a school teacher and my first math tutor. My parents fostered it and encouraged me at all times in this journey. I owe a great deal to my parents and my brothers. But for their love and support, my achievements would have been impossible. I am deeply indebted to my wife Jaya for her love and strong support of my academic pursuits. Ever since we met, she has been a positive influence and a constant source of inspiration in my life. This thesis is dedicated to her. Finally, thanks to my daughter, Amrutha, for adding so much pleasure in my life and allowing me spend extended hours on my research and dissertation.

Contents

List of Tables

List of Figures

Abstract

Statistical Language Modeling has recently been applied to the problem of Information Retrieval (IR) and appreciable performance improvements have been achieved over traditional vector space and probabilistic retrieval models. Most experiments demonstrating the language modeling approach to text retrieval have been based on smoothed unigram language models that only exploit the term occurrence statistic in probability estimation. Experiments with additional features like bigrams have met with limited success. However, language models incorporating n-gram, word-triggers, topic of discourse, syntactic and semantic features have shown significant improvements in speech recognition.

The main thrust of this dissertation is to identify the need to *design* language models for IR that satisfy its specific modeling requirements and demonstrate it by designing language models that (1) incorporate IR-specific features (biterm language model), (2) correspond to better document and query representations (concept language model) and (3) combine evidence from the different information sources (language features) towards modeling the relevance of a document to a given query (maximum entropy language models for IR).

Prior research in incorporating additional features in language models for information retrieval have adopted models derived for speech recognition. However, speech recognition and information retrieval have different language modeling requirements. For example, in speech recognition the utterances *information retrieval* and *retrieval of information* should be distinguished; whereas they should have same representation for efficient

information retrieval. *Biterm Language Models* – a variant of bigram language models – are introduced here to address this problem.

Biterm language models handle the local variation - within 2 words - in the surface form of words used in the expression of a concept. It is, however, these concepts that need to be modeled in the queries to improve retrieval performance. *Concept Language Models* (CLM) that prescribe a two-layered query model is presented in this dissertation. A user's information need is modeled as a sequence of concepts and the query is viewed as an expression of such concepts of interest. It is shown that such a model provides significant improvements to retrieval performance. CLM also provide a natural way of incorporating concept synonymy in the language modeling approach to IR.

Mixture models combine statistical evidence from different sources to estimate the probability distribution. For example, smoothed unigram language model for a document combines unigram counts from the document and corpus. While mixture models are easy to implement, they seem to make suboptimal use of their components. A natural method of combining information sources based on the Maximum Entropy Principle has been shown to contribute significantly to perplexity reduction and hence better language models for speech recognition. Such a framework is proposed for information retrieval in the context of document likelihood or relevance language models. The *maximum entropy language model* for information retrieval provides a better mechanism for incorporating external knowledge and additional syntactic and semantic features of the language in language models for IR.

The dissertation includes empirical results of retrieval experiments using the proposed language models on standard document collections.

Chapter 1

Introduction

My thesis demonstrates that language features beyond term occurrence statistics contribute to the estimation of probability of relevance of documents to a given query. Retrieval models can be *designed* to effectively incorporate such information to achieve improved performance. In a language modeling approach to information retrieval, this corresponds to *designing* better, feature-rich language models for information retrieval.

Significant amount of information is available in the form of digital documents, mostly with textual content, on the world-wide-web as well as in specialized document collections. Web search engines have indexed billion of documents on the web. As of the year 2001, the Google [1] index was reported to have more than 3 billion documents, including 2 billion web pages [67]. In June 2004, the Google index included 4.3 billion web pages. The *information overload* problem, i.e. the inability to extract needed knowledge from an immense quantity of information [81], partly due to large size of the content in the case of web, is also observed in specialized or domain-specific document collections.

[1] `http://www.google.com`

While web search engines return satisfactory results to keyword queries, performance of retrieval systems on specialized collections like TREC [50], MEDLINE [51] with queries representing well-defined information need still has scope for improvement. An example of query with precise information need in TREC is *How often were the peace talks in Ireland delayed or disrupted as a result of acts of violence* (TREC8 topic 404).

While fast retrieval time is a key parameter for web searches, to get precise or better results, the user may be willing to give some more time to the search system to analyze the query and obtain better representation for their information need. A significant difference between web documents and specialized document collections (at least those considered in this dissertation) is the absence of hyperlinks between documents. Web search engines utilize these interconnections between documents to improve their retrieval performance. The dissertation focuses on representing user queries using features beyond term occurrences and propose methods for designing language models for retrieval that exploit such features.

Given a document collection, the task of information retrieval is to identify a ranked list of documents from the collection that satisfy a user's information need expressed in the form of a query. Some aspects of this problem are difficult to model. A clear and precise expression of the user's information need is difficult to capture in query text. Users with two different information needs may generate the same query text. Some computable model is required for relevance or what satisfies the user's information need. Information retrieval systems do not attempt to define the concept of relevance but provide some method for measuring the probability of relevance of a document to given query. Due to this imprecise definition of information retrieval task, performance of retrieval systems are empirically evaluated by a user.

The concept of relevance is an important aspect of information retrieval. The relevance information for a query is usually given by a subset of documents that are *deemed* to satisfy the user's information need. It is usually *topical relevance* that is used as the criteria for identifying relevant documents. It is also assumed that relevance of a document is *independent* of relevance of other documents to a given query. *Adhoc* document retrieval is based on these two assumptions of relevance. The main contributions of this dissertation are for the *adhoc* retrieval problem. Researchers have explored interdependence of relevance of documents [85]. Recent efforts in the HARD track of TREC [1] extend relevance beyond topicality. Additional characterization of user's information need is provided in the form of metadata for the query text. For a given topic the user may be interested only in the background of the topic and the retrieval system should rank documents based on their topical relevance as well as their coverage of background information on the topic. Some aspects of my initial effort [74] to solve the HARD problem are presented in Chapters 4 and 5.

With no formal definition, but an approximate model of relevance, most retrieval system adopt the Probability Ranking Principle [59] for identifying relevant documents. The principle suggests that the ranking of documents be based on their probability of relevance. IR systems compute a relevance value for each document and rank documents in the decreasing order of their relevance value. Relevance value estimation is tied to the representation adopted for documents and queries. Words and/or phrases that occur in documents and queries are typically used as elements, also referred to as terms or features, in their representation. Different models have been proposed for representing and retrieving documents. Formal approaches, grouped under the category of probabilistic retrieval models [18], attempt to model the underlying retrieval process and provide probabilistic interpretation of relevance. Empirical models, the variants of the Vector Space Model (VSM) [65], have shown that significant retrieval performance is possible with simple representation and similarity measures. The OKAPI system [61] developed based on the 2-Poisson retrieval models have shown that formal models of retrieval can achieve appreciable empirical results.

Statistical language models, that have found successful application in a variety of natural language processing tasks including part-of-speech tagging, speech recognition and information extraction, have recently been applied to the information retrieval problem. Ponte and Croft [56] proposed a language modeling approach to information retrieval which deviated from modeling relevance to modeling the query generation process. Statistical Language models are associated with documents and the likelihood of their language model "generating" the given query is used for ranking. A number of researchers have motivated the language modeling approach through Hidden Markov Models [46], statistical translation [6], linguistic clues [31] and risk minimization framework [37]. With strong foundation of probability theory, the language modeling approach with simple language models, specifically, smoothed unigram language models, have achieved significant empirical success. Prior work in other applications of language models, like speech recognition, have given some important direction for extensions and new models for retrieval, including some models proposed in this dissertation.

With a unified representation for indexing and retrieval aspects of an IR system based on language models, the IR problem has been reduced to that of probability estimation. The initial retrieval model based on language modeling approach and subsequent extensions have been based on smoothed unigram language models. Language models are smoothed to obtain robust probability estimates and different smoothing techniques have been experimented with for information retrieval based on unigram language models [86]. The *term independence* assumption is used to approximate the query-likelihood probability to a product of query-term probabilities based on the unigram language model. While such an independence assumption is far from reality, it does result in good performing information retrieval models. Language models that incorporate term dependence in the form of N-gram with $N > 1$ have been explored for retrieval [46, 69]. However, limited improvements in retrieval performance by models using these additional features have raised questions about their contribution towards estimating relevance.

Ponte [55] attributes the failure of additional features to provide *useful information* towards reducing the *entropy of relevance* as one of the possible reasons and raises the following questions:

Q1 "Have these techniques failed to work because the necessarily smooth retrieval functions have not allowed them to?"

Q2 "Have they failed to work because the information they provide is overcome by estimation error?" or, Q3 "Have they failed to work because there is no new information to be gained from them?"

These are the central questions that are addressed in the dissertation. Complementary work in other applications of statistical language models, specifically, speech recognition, provides some hints to answer these questions.

Addressing these questions in the reverse order, significant performance improvements have been achieved using features like n-grams, word triggers, syntactic and semantic dependencies in speech recognition [62, 80, 36]. Hence, Q3, and with some confidence, Q2 can be answered in the negative. Additional features do have some new information to contribute to for generating the language models and the estimation errors can be overcome. To incorporate such additional features, language models have to be *designed* to address the specific requirements of the information retrieval problem. There are interdependencies among the additional features considered and such dependencies should be recognized and modeled in the smooth retrieval functions.

In this dissertation, retrieval models that use IR-specific features beyond term occurrence are proposed. These features are justified based on their relevance to the information retrieval problem. N-gram models, that has been shown to be very effective for speech recognition, use the previously seen $N - 1$ words to predict the next word. It is important for speech recognition to correctly identify the spoken words in the order of their utterance. Biterm language models [75], proposed in Chapter 3 relaxes this constraint to demonstrate that language models can be *designed* with IR-specific features to improve retrieval performance. Word-order constraint, however, cannot be relaxed in all settings. A *race horse* is different from a *horse race*. The dependencies between query terms is an important aspect that needs to be captured in retrieval models. Bigram language models can be viewed to make the assumption that the current word is the best trigger for the next word. Maximum bigram language models [77], proposed in Section 3.2, relaxes this dependency on previous word and selects the best trigger from the query terms on a document-by-document basis. Biterm language models and maximum bigram language models, by using IR-specific features for query and document representations, incorporate local statistical based dependencies in query likelihood models.

In the setting of query-likelihood models, a family of language models named concept language models [76] is proposed here as a new approach to the query generation process. A user query is a expression of user's information need. It is not the words in the query text but the concepts they refer to that is of interest to the user. A two-layer language modeling approach is proposed where a query is viewed as a sequence of concepts and each concept is expressed in the query by a sequence of words or query terms. The syntactic parse of the query is used to identify concepts in it and documents are ranked based on a two-layered language model of the query. It is noted that only the queries are parsed to identify concepts; proximity constraints are used to identify concepts in documents and estimate concept probabilities.

The bag of words model for document and query representation is used in different retrieval models based on language modeling approaches. However, this statistical approach has limitation on the types of features that can be elegantly included in these models. A variety of features, including specific preferences of a user, contribute to the relevance of a document to the user's query. For example, in search personalization, a user's profile could identify specific interest in *fly-fishing* and retrieval results for the user for queries about *fish* should give more weight to documents about *fly-fishing* or *fishing* as a sport. In the presence of topic information about documents, a simple solution is to filter or reweight retrieval results to satisfy the user's preferences. In risk-minimization retrieval framework, Lafferty and Zhai [37] suggest that the query model can encode detailed knowledge and the context in which a user makes the query and loss function can be used to enforce user preferences during retrieval. This dissertation proposes maximum entropy language models that incorporate diverse set of features in a formal way to predict likelihood of a document being relevant to a given query.

Language models that incorporate syntactic and semantic features have been explored for speech recognition and have resulted in significant improvements over N-gram models. Identifying the difficulty in including such information in language models for speech recognition, Rosenfeld [63] proposed a whole sentence maximum entropy language model that views a sentence as a "bag of features" and predicts the sentence probability directly rather than as a product of constituent word probabilities. A similar approach is proposed in Chapter 5 for information retrieval. A relevance model for the given query is built using the query features and documents are ranked based on their likelihood of being generated by the relevance model.

From a natural language processing perspective, the contributions of this dissertation can be viewed as a progression from word or term occurrence based statistical models to phrase or term co-occurrence based models to more semantics-based (at sentence or document level) models. A term's contribution to query or document representation in term occurrence based models is restricted to the variability in its occurrence statistics found in the document or corpus. A number of functions of the document and corpus frequency have been used in vector space and probabilistic models to achieve substantial retrieval performance. Since the semantics of the terms is loosely captured by the occurrence statistics, term co-occurrence based models have been explored to exploit the context of term occurrence to capture semantics. Biterm, maximum bigram language models and concept language models capture the query semantics using term co-occurrence. These models, however, are *defined* based on the specific requirements of information retrieval. Concept language models capture the query semantics using a two-layered model: the first layer views the query in terms of "concepts" of interest to the user and the second layer maps the "concepts" to their surface forms as found in the query or a document. The maximum entropy language models attempt to go beyond the phrase level representation of information need and capture sentence or document level semantics. Non-textual features like topicality of query, number of query term matches in a document, user's preference for particular aspect or sense of a query term contribute to the relevance of a document to the given query. These capture query semantics beyond what is expressed in the words and phrases of the query text. The progression from pure term occurrence statistics based models to semantic models is emphasized in this dissertation.

By introducing new IR-specific features, a new approach for modeling the querying process and new model for incorporating diverse features towards estimating relevance of a document to given query, the dissertation identifies the importance of *designing* language models targeted towards information retrieval and illustrates such models do result in improved retrieval performance through empirical results.

Chapter 2

Background

This chapter presents background and related work in information retrieval and language models. Two widely used class of retrieval models – vector space models and probabilistic retrieval models – are discussed in Section 2.1. This dissertation focuses on methods to incorporate richer features in a particular probabilistic retrieval method that is based on statistical language models. Statistical language modeling is presented in section 2.2.1 followed by the language modeling approach to information retrieval. Depending on the representation and ranking function selected, language modeling approach to information retrieval can be classified into three categories: Query-likelihood models (section 2.2.2.1), Document-likelihood models (section 2.2.2.2) and Query-Document Model Similarity (section 2.2.2.3). The models proposed in this dissertation are based on Query-likelihood and Document-likelihood approaches.

Complementary work in different applications of language models, including speech recognition, provide direction for improvements to the retrieval models. While the language modeling requirements of speech recognition differ from that of information retrieval, prior work in speech recognition have lead to advances in information retrieval. Some of the approaches proposed in this dissertation are motivated by similar techniques in speech recognition. Section 2.3 presents the application of statistical language models to the speech recognition problem and the maximum entropy language models for speech recognition that can incorporate diverse information sources for estimating the word-prediction probability.

In N-gram and maximum entropy language models for speech recognition, the probability of an utterance or sentence is estimated as a product of the word-probabilities given their context. In whole sentence maximum entropy models, discussed in section 2.3.2, the sentence probability is directly estimated by viewing a sentence as a “bag of features”. A similar approach is proposed in Chapter 5 for document retrieval.

2.1 Retrieval Models

This section starts with a discussion of vector space models followed by probabilistic retrieval models. Crestani et al. [18] group probabilistic models into *model-oriented* and *description-oriented* approaches. Section 2.1.2 covers the probabilistic models. Language modeling approaches to information retrieval, that falls under probabilistic retrieval models, are the focus of this dissertation and section 2.2.2 presents prior and related work using this approach.

2.1.1 Vector Space Model

In vector space model for information retrieval [65], documents and query are represented as weighted vectors in an N-dimensional space where N corresponds to the number of terms in the collection. Documents are ranked based on the "distance" of their vectors from the query vector. Different distance measures and weighting schemes have been experimented with over the years with the cosine correlation measure and the TF-IDF (Term Frequency-Inverse Document Frequency) based weighting scheme being the preferred distance measure and weighting scheme, respectively, employed in retrieval systems.

The vector space model provides a good abstraction for the retrieval process. However, it leaves unspecified the particulars of the terms used, the weighting scheme employed for the terms, and the distance measure used to compare document and query vectors. The method of assigning index terms to documents is not related to the method employed to retrieve relevant documents. The only connecting concept is that of representation in a vector space. Relevance feedback is a mechanism where documents deemed relevant to a given query are used to refine the query and, possibly, result in improved retrieval. While different methods for incorporating relevance feedback in vector space models have been explored, they have been added to the vector space model on an ad hoc basis. The model definition lacks guidelines for improving the retrieval performance.

Multi-word terms, like phrases, have been explored for representing documents and queries. Each multi-word term is identified by a particular dimension in the N-dimensional vector space. Statistical and syntax based methods to identify phrases and different weighting schemes have been explored in the settings of vector space models. Different researchers have found phrases as good indexing terms [47, 49] however some approximations were required to handle longer phrases. The correlation between a phrase and its constituent words are associated with different vector dimensions. Their correlation is identified by weighting the phrases and its constituent terms appropriately. Language modeling approach provides a formal method for incorporating such term dependencies and these are explored in this dissertation.

2.1.2 Probabilistic Retrieval Models

Based on the principles of probability theory, the probabilistic retrieval models explicitly incorporate the notion of relevance of a document to given query. The Probability Ranking Principle [59] is applied in most probabilistic retrieval models. It suggests the ranking of documents be based on the probability of their "relevance" to user's information need. Different methods have been proposed to model relevance and estimate the probability of relevance. The language modeling approaches (Section 2.2.2), however, do not explicitly model relevance. The initial language modeling-based approach to information retrieval did not refer to the concept of relevance.

The concepts, *relevance* and *probability of relevance*, are loosely defined in information retrieval theory. Different probabilistic models use different definitions of relevance to model the retrieval process. While relevance is subjective and largely based on a user and their precise information need, relevance of a document to a given query is modeled as a binary random variable, R: A document is either relevant or not relevant to a given user query.

Probabilistic retrieval models differ in the method employed in estimating the probability, $P(R|q, d_j)$, of relevance of each document d_j in the document collection for a given query q. Crestani et al. [18] broadly classified probabilistic retrieval models into two approaches: Model-oriented approaches and Description-oriented approaches.

The *model-oriented approaches* make some probabilistic assumptions on the elements (or terms) used in representing documents and queries. Some probabilistic independence assumptions are made in estimating the probability of these representing elements and these probabilities, in turn, are used in estimating the relevance of documents. In Binary Independence Retrieval (BIR) model [58] proposed by Robertson and Sparck Jones documents are ranked based on the likelihood ratio ($P(d|R)/P(d|\tilde{R})$) of their relevance to the given query. Assuming term independence, the term probabilities are estimated based on the relevance information available for the query. The term weights used in these models are given by

$$w_t = log(\frac{P(t|R) \times (1 - P(t|\tilde{R}))}{(1 - P(t|R)) \times P(t|\tilde{R})} \tag{2.1}$$

where $P(t|R)$ ($P(t|\tilde{R})$) is the probability of term t being present in a relevant (respectively, non-relevant) document. Croft and Harper [19] proposed alternate methods for estimating the term probabilities without the relevance information. In the absence of relevance information, the term probability in relevant documents is assumed to be a fixed probability value (usually, set to 0.5). The probability of a term appearing in a non-relevant document is set to the ratio of documents in the collection with the term.

While BIR models learn model parameters from relevant documents for a given query, Binary Independence Indexing (BII) models [43] learn the parameters for a document in relation to a number of queries. In indexing models the objective is to identify if a term can be associated with a given document and quantify the term's contribution to the document. The term contribution is based on the document's relevance to queries using or containing that term. Learning the indexing weight for terms in a document from prior queries, the model is applied to future queries.

Parametric methods like 2-Poisson [8] and N-Poisson [42] models are also model-oriented approaches. In 2-Poisson models it is assumed that the number of occurrences of a term in a document depends on the relevance of the document and a Poisson distribution is used to model the occurrence distribution. In N-Poisson extends the 2-Poisson model from two class (relevant and non-relevant documents) to partitioning of the document collection to N classes. The N classes can be viewed as topics in the collection and it is assumed that the occurrence distribution of a term varies across topics.

In BII models the probability of relevance of a document given a query is computed relative to evidence consisting of the properties of queries for which the document was considered relevant. Where as, in BIR models, the probability of relevance is computed relative to evidence from documents considered relevant to the given query. Robertson et al. [60] proposed Model 3, a unified model that combines the retrieval strategies of BII and BIR models. Fuhr's *retrieval with probabilistic indexing* model [24] explored Model 3 by using the following retrieval function:

$$r(d_m, q_k) = \sum_i w_m i \cdot v_k i \tag{2.2}$$

where $w_m i$ is the weight assigned to an indexing term,t_i, for document, d_m, by a probabilistic indexing model and $v_k i$ is the weight of query term, t_i for given query, q_k.

Description-oriented approaches associate a set of features or descriptors for query-document pairs and use these features to estimate the probability of relevance. A training set of query-document pairs, typically consisting of prior queries and their relevant documents, is used in the estimation process. Examples include Fuhr model [24, 25] and Staged Logistic Regression [17].

In Fuhr's model, the probability of relevance, $P(R|q_k, d_m)$ is estimated from relevance data in the form of query-document or term-document pairs. The model associates *relevance descriptors* or set of attributes or features, $s(t, d_m)$, to a term-document pair (t, d_m) and estimates the probability of relevance as $P(R|s(t, d_m))$. Two or more term-document pairs can share the same relevance description s leading to better estimation of the probabilities with limited data. The relevance descriptors can be any attribute of the term-document pair like frequency of occurrence of the term in document, information about location of occurrence of term in document. Different indexing functions that use linear regression [25] and, later, logistic regression [26, 27] have been experimented with.

The initial estimation method in Fuhr model was based on linear regression. Due to the binary nature of the dependent variable – relevance or non-relevance of a document – Cooper criticized the use of linear regression that is typically used when dependent variables are drawn from a continuum of values in the above and suggested logistic regression to a more appropriate one. Cooper et al. [17] proposed a description oriented approach using logistic regression. That staged logistic regression model treats the set of attributes or relevance descriptors as clues for relevance of a document to given query. These clues can be complex in the sense that a clue can be a composite of multiple simple clues. The occurrence of a query term in a document is a composite clue that is supported statistically by a number of simpler clues: its relative frequency in the document, its frequency in the given query, inverse frequency in the collection, etc. The two stage logistic regression model ranks documents based on the log odds of relevance given by

$$\log O(R|a_1, a_2, \ldots, a_n) = \log O(R) + \sum_{i=1}^{n}(\log O(R|a_i) - \log O(R)) \tag{2.3}$$

where $\{a_1, a_2, \ldots, a_n\}$ are the set of complex attributes. The *linked dependence* assumption given by

$$\frac{P(a,b|R)}{P(a,b|\tilde{R})} = \frac{P(a|R)}{P(a|\tilde{R})} \cdot \frac{P(b|R)}{P(b|\tilde{R})} \tag{2.4}$$

where a and b are attributes, is used in estimating (2.3). The second stage of the model views each complex attribute as a set of elementary clues and estimates the log odds using logistic regression.

Using a set of attributes or features rather than the index terms themselves for estimating probability of relevance has some advantages. Limited availability of training data is a problem for BII models (queries for which the current document is relevant). The use of descriptors reduces the required training data and also facilitates collection-specific tuning of the parameters. Most model oriented approaches are tightly coupled with the form of representation of documents and queries. Hence, any modification to the form of representation requires a new model to be developed. Description oriented approaches provide the independence from particular form of representation. Selecting a good set of attributes or features is a problem that needs to be tackled in description oriented approaches.

Language modeling approaches to information retrieval (ref. Section 2.2.2) also encounter similar problem of requiring to develop new models to incorporate changes to document and query representations. The maximum entropy language modeling method proposed here provides similar "representation independence" to language modeling approaches to information retrieval.

Probabilistic retrieval models discussed above are tightly coupled with the document collection. Relevance for a query is defined based on the documents in the collection and the parameters learned for a collection will not necessarily work well for a different document collection. Models based on uncertain inference provide a means to learn the probability of relevance, $P(R|q_k, d_m, K)$, of a document d_m to query q_k given some prior knowledge, K. The prior knowledge can include domain of document collection and queries, knowledge about the user and relevance feedback. In inference network models, the probability of relevance is estimated by the inference process of finding and evaluating a relation between the document and query. INQUERY [10] is a Bayesian inference network system for information retrieval. Nodes represent the entities in an IR system: documents, index terms, concepts, queries, and information need. Edges represent probabilistic dependencies between them. In Bayesian networks, the probability of an entity being true is estimated based on the probability of its parents in the inference network being true.

The inference network for information retrieval has two component networks. The document network represents each document in the collection related to the terms that occur in it. A query network, generated for each information need, is related to the static document network in order to process the query. Given query, q_k, the probability of relevance of a document, d_m is estimated by $P(q_k = true|d_m = true)$. Different retrieval models can be shown to be special cases of the inference network model. It is also possible to use multiple representations for documents and queries to effectively combine evidences from these representations towards measuring probability of relevance. However, the restriction of evidence (node values) being binary (feature is present or absent) limits the use of inference network model for IR when certain types of evidence are available.

While the probabilistic models based on the principles of probability theory provide a formal model for indexing and retrieval process, there is scope for efficient and robust estimation techniques for the probability of relevance. Some simplification assumptions are made to address the modeling and estimation issues. Indexing models associate a term with a document if the document will satisfy a user when the user's query is that term. Ponte and Croft [56] while proposing their language modeling approach to IR identify the parametric assumptions on term distributions, as in 2- and N-Poisson models, are unnecessary since the data (document collection) whose characteristics are being modeled is already available. Non-parametric methods are central to language modeling approach discussed later in Section 2.2.2.

The linked dependence assumption (2.4) is widely used in probabilistic retrieval models discussed above as well as in language modeling approaches (ref. Section 2.2.2). While description-oriented approaches can handle any set of computable features, not necessarily independent in the relevance estimation, most experiments have used only the statistical properties of term occurrences as features. The maximum entropy model proposed here can combine information from diverse information sources of relevance that can be based of statistical and semantic properties of documents and queries.

2.2 Information Retrieval using Statistical Language Models

2.2.1 Statistical Language Models

A language model is a probability distribution defined over strings of a language. In the case of natural language, the probability distribution is over set of strings. The typical usage of language models is for predicting the probability of a particular set of strings. In some applications, the sequential aspects of the strings (or word sequences) may play a role in the estimation of the probability distribution. For example, a speech recognizer should identify words in the same order they were spoken by a speaker. The next utterance usually depends on the previous spoken words. In information retrieval, unigram language models use words in the representation and ignore the sequential aspects or word order for probability estimation. Language models can be designed to capture particular characteristics of the application. Robust probability estimation is important for precise prediction and smoothing techniques are used to obtain robust estimates in most applications.

Smoothing is an important aspect of estimating the probability, $P(X = x)$, a real valued random variable X takes on a value of x. The estimation can be based on counting occurrences in some training data or observations of the random variable X. Not enough data is available, even in the discrete case, to estimate by counts. The errors made in probability estimation from training data are analyzed in the context of bias-variance tradeoff. A bias error is observed if the estimation is not close enough to the training data. An estimation with variance error fits the data too well; also referred to as overfitting.

In estimating the probability distribution from some training data, it is possible to make a parametric assumption of underlying distribution of the data and the parameters can be estimated from the training data. For example, a Gaussian assumption of the data would require the mean and variance of the distribution learned from the training data and the probability distribution is given by a normal distribution with the estimate parameters. While the distribution so estimated will be a good fit if the data indeed is Gaussian, otherwise, the estimate suffers from bias error.

Estimating the probabilities from counts introduces variance error. An unseen value would be given a zero probability, resulting in incorrect estimation on novel data. However, non-parametric approaches based on occurrence counts is preferred in language modeling since it lets the data "speak for itself". Variance error can be handled, to some extent, via smoothing of the estimates. A number of smoothing techniques have been explored and compared in the context of language modeling for IR [86]. Specific methods used in this dissertation are presented later in this chapter.

2.2.2 Language Modeling Approach to Information Retrieval

Probabilistic retrieval models focus on the concept of relevance and estimate $P(R|d, q)$ the probability of relevance given document and query. While different methods have been proposed for modeling relevance, some parametric assumption of term distribution or relevance is employed in probabilistic retrieval models. Ponte and Croft [56] first proposed a probabilistic approach to information retrieval that is based on statistical language models and did not explicitly use the concept of relevance. Each document is assumed to be generated by a stochastic process and they are deemed relevant to a given query if the query is likely to be have been generated by the same stochastic process. Subsequent publications have motivated this idea through various perspectives (HMM,statistical translation, risk minimization) and have also proposed alternative formalisms for ranking documents. In addition to *ad-hoc* retrieval, these approaches have also proven to be effective in a number of IR applications, including question-answering [5], text segmentation [4], and summarization [7]. A significant aspect of language modeling approach is that it provides unified model that can be used for both indexing (document representation) and retrieval. In addition, the model includes the mechanism to weight or rank documents for a given query.

Based on the document representation and ranking, language modeling approach to information retrieval can be categorized in to three classes as illustrated in Figure 2.1. A document can be viewed as an outcome of the choice of words and constructs the author makes in writing the document. A document or author model can be associated with each document. A user's query generation process is modeled as the user selecting the query text from the features of an "ideal" document that the user believes will satisfy her information need. A query model can be associated with the user's information need representing the characteristics of an "ideal" document. The language modeling approach attempts to pre-

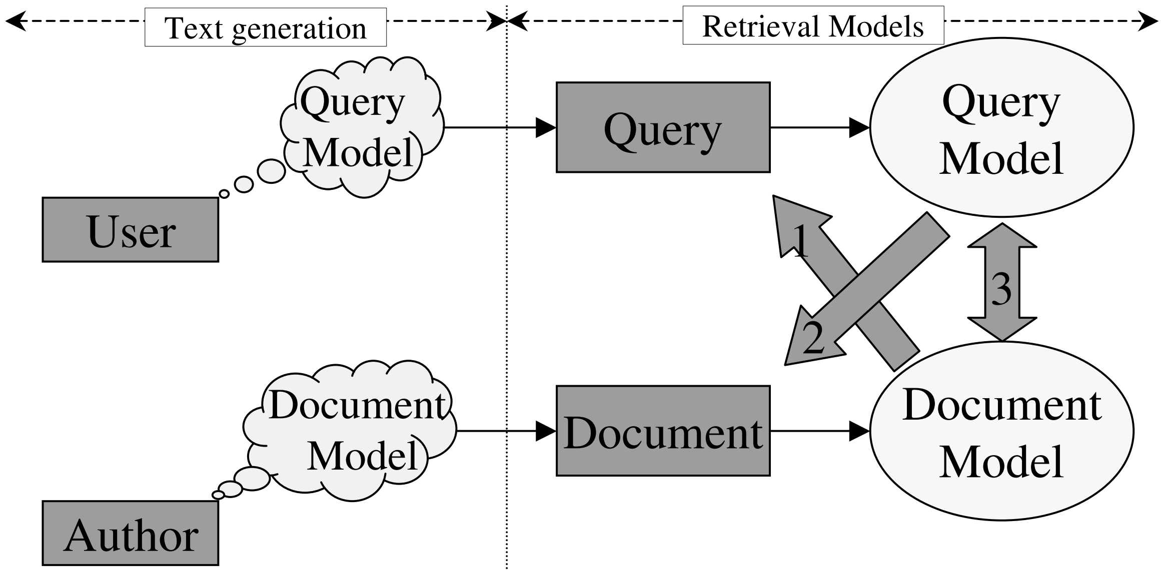

Figure 2.1: Document representation and ranking options in Language Modeling approaches to IR

dict either or both of the document and query models. The following three retrieval models are identified in Figure 2.1:

1. *Query-likelihood Models*: A language model is associated with each document and the documents are ranked based on the likelihood of their model generating the given query.

2. *Document-likelihood Models*: A model of the user's information need in the form of a query model is generate and documents are ranked based on the likelihood of their content being generated by the query model.

3. *Query-Document Model Similarity*: Language models are associated with both query and documents and the model similarity is used to rank documents.

More details about these models are presented in the following sections.

2.2.2.1 Query-Likelihood Models

Ponte and Croft [56] first proposed the language modeling approach for information retrieval: Each document D is viewed to have its own language model M_D. Given a query Q, document D is ranked based on the probability $P(Q|M_D)$ of the query being *generated* by its language model. Assuming term independence and viewing query as a set of terms, Ponte and Croft [56] computed the query-likelihood probability using *multi-variate Bernoulli* model:

$$P(Q|M_D) = \prod_{w \in Q} P(w|M_D) \prod_{w \notin Q} (1 - P(w|M_D)). \tag{2.5}$$

The probability estimation takes into account the presence of query terms and absence of other terms in the document. This is similar to Robertson-Sparck Jones' Binary independence Retrieval model. A non-parametric estimate is used for the term probability, $P(w|M_D)$. For terms occurring in the document the estimate is computed based on the average probability of w in documents containing it and a risk factor corresponding to the use of the average probability instead of the exact maximum likelihood estimate. The estimate for non-occurring term is given by the corpus probability, cf_t/cs – the ratio of the total number of occurrences of the termcf_t, in the collection to the collection size, cs. Ponte and Croft [56] empirically showed the language modeling approach to give statistically significant improvements over vector space model using tf-idf/OKAPI weighting [61] (around 20% improvement in average precision on TREC queries 202-250 on TREC disks 2 and 3).

Hiemstra [30], Miller et al. [46] and Song and Croft [69] used a multinomial model of query generation where one is interested not in the presence or absence of a word but the identity of the word in a particular position. Given a query $Q = q_1, q_2, \ldots, q_m$, the probability of generating the query estimated based on the term q_i being selected as the ith word in the query. Making the term independence assumption, the query-likelihood probability is given by

$$P(Q|M_D) = \prod_{i=1}^{m} P(q_i|M_D). \tag{2.6}$$

A smoothed unigram model is used for the query term probability; Specifically, a linear interpolation, also referred to as Jelinek-Mercer smoothing, of maximum likelihood estimate of the term document and the probability of the term in the collection:

$$P(w|M_D) = \lambda P_{ml}(w|M_D) + (1 - \lambda)P_C(w) \tag{2.7}$$

where λ is the interpolation weight between 0 and 1. Hiemstra [30] uses document hits to estimate collection probability as

$$P_C(w) = \frac{df_w}{\sum_v df_v} \tag{2.8}$$

where df_w is the number of documents in which the term w appears.

Miller et al. [46] arrived at the query-likelihood probability in (2.6) using a 2-state Hidden Markov Model (HMM) for query generation given in Figure 2.2. The two states correspond to the document language model and the general English language model. At the point of selection of the next query term, the model first selects the state and then a word from the model corresponding to that state. The model selects the "document" state with probability λ and the "General English" state with the probability $1 - \lambda$. The term probability in "Document" state the term frequency in the document. In the "General English" state, the term probability is given by its corpus probability. Miller et al [46] also showed

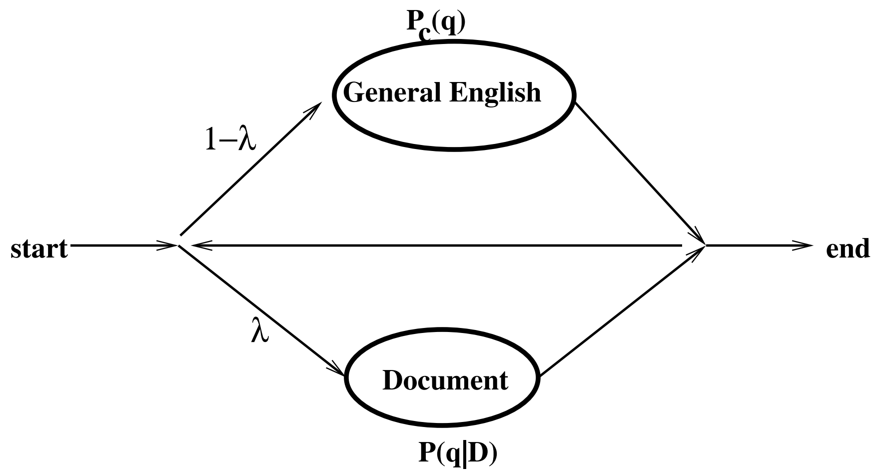

Figure 2.2: Simple two-state HMM for Query-Likelihood Language Model for IR

that their base model using HMMs can be extended to incorporate other word-generation mechanisms such as N-grams, synonyms, and topic lexicons. Song and Croft [69] use the sequence view of queries and estimate term probabilities using Good-Turing smoothing.

Berger and Lafferty [6] proposed a statistical translation based approach to information retrieval where the query generation from information need is modeled as a "noisy channel" of communication. The relevance of a document is measured by the probability that the query would have been generated as a "translation" of the document. The query likelihood probability is estimated as

$$P(q|d) = \prod_{i=1}^{m} \sum_{w} t(q_i|w)p(w|d) \tag{2.9}$$

where the translation model, $t(q|w)$, maps a term w in a document to a query term, q. Synonymy and polysemy are important issues in information retrieval. They correspond to the problem of multiple terms having the same meaning (example, *pope* and *pontiff*) and more than one meaning is attributed to the same word (example, *bank* as in financial bank and river bank). The translation model performs some kind of semantic smoothing to handle the synonymy and polysemy problems. While significant improvements have been obtained using this model, it requires sufficient training data (query-relevant documents pair) to learn the translation probabilities. Berger and Lafferty generate synthetic data to estimate these probabilities. The context of term is important to get its semantics in a document. The translation model, being a context-independent model, is unable to incorporate context information in the model. A large number translation probabilities have to be estimated from the training data to support higher-order document representations. Added to this is the complexity of the summation in Equation (2.9) being over all words in the document.

A number of smoothing techniques have been experimented with for probability estimation in the context of query-likelihood models. Zhai and Lafferty [86] compared linear interpolation (also known as Jelinek-Mercer smoothing), Dirichlet priors and Absolute smoothing techniques for information retrieval over different data sets. The linear interpolation method of smoothing is given in Equation (2.7). The probability of a term in a document is estimated by a weighted combination of the empirical probability of selecting the term from the document and the empirical probability of selecting the term from the document collection. The weights can either be set manually based on the “modeler”’s notion of importance of the document statistic over the collection statistic or can be learned from some training data. In Bayesian smoothing using Dirichlet priors, the language model is assumed to be multinomial with the conjugate prior for Bayesian analysis as the Dirichlet distribution $\{\mu P_C(w_i)\}$. The Dirichlet prior smoothed term probability is given by

$$P(w|M_D) = \frac{n(w,d) + \mu p_C(w)}{\sum_v n(v,d) + \mu} \tag{2.10}$$

where μ is the Dirichlet prior parameter.

While the above discussion of language modeling approaches for information retrieval were based on unigram language models, higher order models incorporating the context of the words have been explored and such models are discussed in section 2.2.2.4. This dissertation proposes query-likelihood models that incorporate document features beyond term occurrences.

2.2.2.2 Document-Likelihood Models

Classical probabilistic retrieval models are based on the *probability ranking principle* [59] that suggests documents be ranked based on their probability of being *relevant* to the information need of the user. These models provide intuitive interpretation to the concepts of relevance feedback and automatic query expansion used in IR. In an attempt to explicitly include the concept of relevance in language modeling, Lavrenko and Croft [39] introduced relevance-based language models which assumed the distribution of terms in the topic of interest is known and computed the probability of relevance for a given document based on the probability of the document being generated by the topic model. The topic model was generated using the top N documents returned by an initial search using the user's query. In their experiments they used the smoothed bigram-unigram model proposed by Song and Croft [69] to obtain the top N documents. The relevance-based language modeling for IR is discussed further in chapter 5 where the maximum entropy language models are proposed for estimating the relevance models.

2.2.2.3 Query-Document Model Similarity

Vector space retrieval model is a similarity based approach that uses a vector representation for documents and queries and uses the similarity between their vectors to rank documents. In their risk-minimization framework that is rooted in Bayes decision theory, Lafferty and Zhai [37] propose model-similarity based method for information retrieval. Language models are associated with documents and queries and the similarity or distance between these models is used as the relevance status value for the document. Depending on the particular selection of the risk or loss function, different retrieval models can be shown to be special cases of the risk-minimization framework. By associating language models for both queries and documents, this approach can be viewed as a method that combines some of the aspects of query-likelihood and document-likelihood models. Document language models can precisely capture the document generation process and encode the author preferences. User preferences and needs can be encoded in the query model. The closeness of these models is an indicator of the document's relevance to the given query. Smoothed unigram language models have been explored for representing the document and query models. Lafferty and Zhai [37] also propose a Markov-chain based method for generating query models with relevance feedback information.

The higher-order language models explored in this dissertation are based on query-likelihood and document-likelihood models. It is possible to use the model similarity method for higher-order language models, however, appropriate loss functions have to be defined. While the Kullback-Leibler measure can be used to measure the distance between the higher-order models in a larger event space, its implication and relevance to the information retrieval problem needs to be investigated. This is beyond the scope of this dissertation.

2.2.2.4 Higher-order Language Models

For clarity of presentation, all language modeling approaches that use document or query representation beyond term occurrence, like N-grams, phrases, are grouped under higher-order language models. Such features have been used in vector space and classical probabilistic retrieval models, as briefly mentioned earlier in this chapter, and have been shown to improve retrieval performance in certain settings. Section 4.1 discusses prior work in using statistical and NLP techniques to incorporate phrases in retrieval models. The following concentrates on using such features in language modeling approaches.

Miller *et al.* [46] incorporate bigram statistics in their HMM framework by proposing a 3-state HMM that corresponds to a mixture model based on bigram, unigram and corpus probabilities. Song and Croft [69] proposed a general language model that combines bigram language model with Good-Turning estimates and corpus based smoothing of unigram probabilities. Additional features that have been explored for information retrieval – n-grams – were based on experiences from speech recognition. This dissertation explores features that capture IR specific characteristics and incorporate them in language modeling approaches to information retrieval. Biterm language models (see section 3.1) captures one such feature.

Smoothing is essential for higher-order language models due to sparseness of the occurrence of higher-order features in the data. In prior language modeling experiments for IR, mixture models combining unigram and bigram statistics from document and corpus were used for probability estimation. Linear interpolation is used as a smoothing mechanism to "flatten" distributions and assign non-zero probabilities for unseen events. It can also be viewed as a mechanism to combine different information sources where the corresponding weights can either be assigned manually or learned using an Expectation-Maximization type algorithm [23] with some training data. While mixture models are easy to implement and analyze, they make sub-optimal use of their components and are inconsistent with their components [62]. In backoff method [35] for combining information sources, information sources are ordered based on the detail of description they provide on the underlying process of interest and the "best" description is selected in order for a particular instance. Zhai and Lafferty [86] experimented with different smoothing techniques for unigram language models and reported linear interpolation smoothing to perform better than backoff methods for information retrieval.

Only statistical phrases have thus far been used in higher-order language models for information retrieval. The concept language models, proposed in Chapter 4 uses syntactic parse of the query to generate a two-layer language model for retrieval.

Language models, in theory, facilitate rich representations of documents and queries. However, performance improvement has been marginal or non-existent thus far in information retrieval using N-gram features. Characterizing the methods employed on the language modeling approaches to information retrieval as smoothing, Ponte [55] questioned if the smooth retrieval functions have not allowed the additional features to contribute towards relevance estimation. In speech recognition, statistical language models that incorporate a variety of features have been successfully used to effect improved speech recognition. Next section briefly discusses the language modeling approach to speech recognition and the maximum entropy method that is able to combine diverse information sources to generate language models for speech recognition. The maximum entropy models for information retrieval proposed in Chapter 5 are motivated by the performance of similar models in speech recognition.

2.3 Language Models for Speech Recognition

Given an acoustic signal, A, for an utterance, the speech recognition problem is to predict the corresponding sequence of words, $s = w_1, w_2, \ldots, w_n$. Bahl et al. [3] use maximum likelihood estimate to predict the sentence given the acoustic model.

$$s^* = \arg\max_s P(s|A) = \arg\max_s P(s)P(A|s) \tag{2.11}$$

where $P(s)$ is the sentence probability and $P(A|s)$ acoustic model given a sentence. The sentence selection is weighted by using its prediction probability, $P(s)$, as a prior to selecting possible acoustic models. A good prior probability will ensure right candidates are selected for the acoustic model resulting in improved recognition performance. One method to estimate sentence probability is by using the chain rule given by

$$P(s) = \prod_{i=1}^{n} P(w_i|w_1, w_2, \ldots, w_{i-1}) \tag{2.12}$$

where the prediction of the next word depends on all previously seen words, also referred to as the history, starting from the current word.

Conventional N-gram models estimate the prediction probability by approximating this history to the last seen N words. A trigram model has been shown to be very effective as a language model yielding to improved speech recognition. Trigram probability estimation is based on maximum likelihood estimates from a large training corpus. Due to large number of candidate trigrams that are possible using words in a large vocabulary, it is likely for some trigrams to be assigned zero likelihood estimate. Some smoothing technique is employed to address the sparse data problem. A backoff model is one such method where in the absence of sufficient evidence for trigram, the weighted version of the bigram estimate is used to approximate the trigram probability. This method can, in turn, be applied to compute the bigram probability, backing off to the unigram estimate. The uniform probability of a word from the vocabulary is the backoff estimate for unigram probability.

A linear interpolation model that combines the three estimates – trigram, bigram, and unigram maximum likelihood estimates – is given by

$$\hat{P}(w_i|w_{i-1}, w_{i-2}) = \lambda_3 P_{ml}(w_i|w_{i-1}, w_{i-2}) + \lambda_2 P_{ml}(w_i|w_{i-1}) + \lambda_1 P_{ml}(w_i) + \lambda_0 1.0/|V| \tag{2.13}$$

where $P_{ml}(\cdot)$ are the maximum likelihood estimates, $|V|$ is the vocabulary size, and λ_i are the parameters that weight the different estimates. An Estimation-Maximization (EM) type algorithm [23] is typically used to determine the combination weights.

Rosenfeld [62] proposed that the different estimates can be viewed as information sources available in the document history to predict the next word and identified an estimation process based on the Maximum Entropy Principle (MaxEnt) [32] as a method for combining these evidences.

2.3.1 Maximum Entropy Models

As a probability estimation technique from given data, the Maximum Entropy Principle suggests the selection of the most uniform distribution that satisfies some constraints defined on the data. This corresponds to selecting the distribution with maximum entropy. The entropy of a probability distribution, $P(x)$ is given by

$$H(P) = -\sum_{x} P(x) \log P(x) \tag{2.14}$$

Probability estimation using MaxEnt can be viewed as a two step process. The first step involves identifying constraints that the predicted probability distribution should satisfy. In the case of speech recognition, these constraints represent the various information sources that contribute to predicting the next word probability. The second step involves estimating the most uniform distribution that satisfies the constraints defined in the first step. A generalization of MaxEnt is called the Minimum Discrimination Principle or Minimum Cross Entropy Principle (MinXEnt) that is applicable in the presence of a prior distribution and constraints. It suggests the selection of the distribution that is closer to the prior and satisfies the given constraints. MinXEnt is MaxEnt when the prior distribution is uniform distribution.

Formally, given a prior $q(x)$ and the expectation,k_j of the constraint functions, $f_i(x)$, the m constraints are given by

$$\sum_x P(x) f_i(x) = k_i \tag{2.15}$$

Given the constraints, the MinXEnt solution is an exponential distribution of the form

$$P(x) = exp(\lambda_0 + \sum_{i=1}^{m} \lambda_i f_i(x)) \tag{2.16}$$

where λ_i is the weight of the feature, f_i and λ_0 is a normalization constant to ensure $P(x)$ is a probability distribution satisfying $\sum_x P(x) = 1$. The maximum entropy solution is unique and is guaranteed to exist.

The maximum entropy optimization problem can be solved using iterative scaling methods. Darroch and Ratcliff [20] proposed the Generalized Iterative Scaling (GIS) that starts with some initial values for the feature weights λ_i^0 to obtain an initial probability estimate

$$P^0(x) = exp(\lambda_0^0 + \sum_{i=1}^{m} \lambda_i^0 f_j(x)). \tag{2.17}$$

Each iteration creates a new estimate for the probability distribution by adjusting or scaling the feature weights such that it is better, in some sense, than the previous estimate. The jth iteration is as follows:

1. Compute the expectations of all the features f_j's given by (2.15) using the current probability estimate function.

2. The λ_i's are update by comparing the computed expectation value and the estimated (from training data) expectation value using

$$\lambda_i^{(j+1)} = \lambda_i^j + \frac{1}{F_i} \log \frac{k_i}{E_{P^j}[f_i]} \tag{2.18}$$

where F_i is the step size that is set to the number of active features.

3. Compute the new probability estimate using the updated feature weights

The iterations are continued until convergence or near-convergence of the estimate.

Darroch and Ratcliff [20] proved the convergence of the optimization algorithm. However, their algorithm required the feature weights, $f_i(x)$ to sum up to a constant M for all x. This requirement can be satisfied by adding a slack variable or feature that ensures the feature weights for any x sums up to M. Della Pietra et al [22] proposed an Improved Iterative Scaling (IIS) algorithm that does not require to the additional constraint of GIS and showed faster convergence by selecting a larger effective step size than GIS. However, an IIS implementation has been shown to take longer time to reach the solution than GIS implementation [41] as it requires more bookkeeping than GIS. All experiments reported in this dissertation involving Maximum Entropy Models were computed using the GIS algorithm.

The parameter estimation in GIS or IIS, based on maximum likelihood estimate, is susceptible to overfitting. In the GIS update rule (2.18) this corresponds to the empirical expectation $k_i = 0$ leading to the parameter value $\lambda_i \rightarrow \infty$. It is required to smooth the maximum entropy model or adjust parameter values away from the maximum likelihood estimates. Different smoothing methods have been experimented with for maximum entropy models [15]. A Gaussian prior for the parameters, λ_is, is introduced and a *posterior* model instead of a maximum entropy model is estimated. The parameters for a maximum entropy model is estimated by maximizing the log-likelihood $L_X(\Lambda)$ of the training data X

$$L_X(\Lambda) = \sum_x q(x) \log P_\Lambda(x). \tag{2.19}$$

With the Gaussian prior, assuming to have diagonal covariance, the objective function is changed to

$$L'_X(\Lambda) = L_X(\Lambda) + \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp(-\frac{\lambda_i^2}{2\sigma_i^2}) \tag{2.20}$$

$$\approx L_X(\Lambda) - \sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma_i^2} \tag{2.21}$$

where σ_i^2 are the variances for the Gaussian distribution. GIS update rule (2.18) can be easily modified to include the Gaussian prior.

The Maximum entropy models have been successfully applied to a number of machine learning problems such as speech recognition, text classification(ref. Section 2.4), text segmentation [4], part-of-speech tagging [57] and named entity tagging [73]. The ability to reconcile the effect of diverse information sources to estimate the probability distribution makes maximum entropy model a viable solution for any probability estimation problem. In this dissertation, maximum entropy models have been proposed to solve the information retrieval problem.

Additional source of information in the document history have been explored in language models. Class-based N-gram models, long distance N-gram models are some examples. Linear combination of these information sources is possible, however, Rosenfeld [62] identified some shortcomings of using linear combination for combining these evidences and proposed the application of MaxEnt to combine evidences from multiple information sources. Maximum Entropy Language Models derived based on the MaxEnt principle are discussed next.

In speech recognition, Rosenfeld [62] estimated the word prediction probability, $P(w|h)$, given the history using MaxEnt principle. Each information source partitions the event space of (h, w) based on some property of the history, h. In the case of bigrams, the event space is partitioned by the last word in the history. It creates equivalence classes of events in the event space. For a particular equivalence class where the history ends with u, the bigram model assigns same probability for each element in that class and the estimate, $K_{v,w}$ is derived from the distribution of the training data in that class. Similar equivalence classes can be defined on the event space based on other information sources and it is possible for multiple information sources to contribute to the prediction of $P(w|h)$. Linear interpolation reconciles the different estimates by weighting them. In the case of Maximum Entropy models, the inconsistency is removed by relaxing the conditions imposed by these information sources. The estimated probability distribution, $\hat{P}(w|h)$ takes the value $K_{v,w}$ *on the average*. This constraint is given by

$$E_{h\text{ends in}_w}[P(w|h)] = K_{v,w}.$$

Each information source is assumed to define a subset of the event space and each subset imposes a constraint on the estimate $P(w|h)$ that it agrees *on average* with a certain statistic, K, of the training data, defined over the subset. The subset is defined by an indicator function, $f_S(h, w)$ that takes value of 1 when $(h, w) \in S$ and 0 otherwise. The constraints, thus, are given by

$$\sum_{(h,w)} P(h,w) f_S(h,w) = K \tag{2.22}$$

Rosenfeld [62] showed that maximum entropy language models lead to significant reduction in perplexity (a measure of number of choices the language model has at each stage of predicting the next word) with a variety of not-necessarily independent information sources. A number of subsequent publications in speech recognition have used the maximum entropy language models to build better language models using syntactic and semantic clues [12, 36] and topic of discourse [80]. However, these methods require complex models and training procedures to incorporate such information.

2.3.2 Whole Sentence Maximum Entropy Models

The use of the chain rule of probabilities (Equation (2.12)) was identified as one of the estimation methods for predicting the sentence corresponding to a speech utterance (Equation (2.11)). Rosenfeld [63] identified that terms like $P(w|h)$, the word probability, given its history, may not be the best way to think about estimating sentence probability, $P(s)$ and proposed an alternate method to directly estimate the sentence probability.

While local dependencies based on neighboring words can be modeled using some approximation of $P(w|h)$, global sentence information such as grammaticality or semantic coherence are awkward to encode in the conditional framework. Some grammatical structure has been captured in conditional exponential models [36] by formulating the terms based on partial parse trees. External influences on the sentence, like the previous sentences or topic of discourse, are also difficult to encode efficiently in the model. Wu and Khudanpur [80] introduced the (predicted) topic as a independent variable in the history to predict the next word. Since word level prediction probabilities are accumulated to estimate sentence probability, systematic bias in estimation is compounded. The N-gram approximations employed to estimate the probability $P(w|h)$ make independence assumptions that not necessarily reflect the reality of spoken text. There are global dependencies within- and across sentences that affect the sentence probability prediction. Some aspect of the utterances like length of a sentence cannot directly be modeled in an estimation based on the chain rule, while such aspects are important in improving recognition performance.

Instead of representing the sentence by a sequence of words and the associated chain rule, Rosenfeld [63] proposed a whole sentence exponential model that treat a sentence as a "bag of features" where the features, $\{f_i\}$, are arbitrary computable properties of a sentence. A whole sentence exponential language model has the form:

$$P(s) = \frac{1}{Z} P_0(s) exp(\sum_i \lambda_i f_i(s)) \tag{2.23}$$

where the $\{\lambda_i\}$'s are the parameters of the model and Z is a normalizing constant that depends only on the parameters. The probability $P_0(s)$ is a prior or initial distribution that can be a uniform distribution or obtained by alternate formalism (example, by using chain rule and conditional distributions like n-grams). An exponential model (any expression of the form (2.23)) is a maximum entropy model when its parameters are chosen such that the distribution satisfies certain linear constraints. The constraints typically specify an expectation value,K_i for each feature $f_i(s)$:

$$E_p f_i = K_i. \tag{2.24}$$

The expectation values are assigned based on expectation value of the feature under the empirical distribution, $\tilde{P}$, of some training sentences $\{s_1, \cdots, s_N\}$. Among the distributions that satisfy the constraints, the exponential distribution that are closer (in the Kullback-Leibler divergence sense) to the prior distribution is assigned to $P(s)$. If the prior distribution is uniform, the resultant distribution is a maximum entropy distribution.

Two computationally infeasible problems are encountered while using whole sentence maximum entropy models: The normalization constant, Z, is given by

$$Z = \sum_s P_0(s) exp(\sum_i \lambda_i f_i(s)) \tag{2.25}$$

where the summation is over all possible sentence, an infeasible enumeration task. A similar infeasibility is observed in the context of using the GIS algorithm to solve the optimization problem. The GIS update rule (2.18) requires the expectation value of the estimated probability distribution. This expectation is computed by summing over the feature values across all possible sentences. Fortunately, both these infeasible task can be avoided when using the whole sentence language models for the purpose of ranking candidate sentences. During training, the evaluation of the expectation value can be achieved by sampling. The normalization constant,Z, is required only when one is interested in computing the perplexity of the language model and it can be ignored in the context of speech recognition where its value will be same for all candidate sentences for the current utterance.

Chen and Rosenfeld [14] compared different sampling techniques for estimating whole sentence maximum entropy language models. Metropolis sampling [45] starts with an initial sentence. For each word position in that sentence, a replacement word is proposed and the changed sentence is accepted with some probability. After iterating through all positions of the sentence, the resultant sentence is added to the sample. The sample generation depends on the distribution used to select the replacement word. In *importance sampling*, a sample of sentences, $\{s_1, \ldots, s_M\}$, generated according to some distribution, $Q(s)$ (a distribution that should be close to $P(s)$ for efficient sampling). For each sentence, the statistic $[\frac{P(s)}{Q(s)} f_i(s)]$ is calculated for each i and their expectations given the desired expectation,$E_P[f_i(s)]$. A variation of the importance sampling is used in the estimating maximum entropy language models for information retrieval in Chapter 5.

Cai et al [9] use whole sentence language models to incorporate semantic coherence of words in a sentence to improve upon a trigram language model for speech recognition. They also provide a different interpretation for the whole sentence maximum entropy models in terms of logistic regression. They define a new random variable Y such that

$$Y = \begin{cases} 1 & \text{if } s \in P \\ 0 & \text{if } s \in P_0 \end{cases}$$

where $s \in P$ means the sentence is from the original corpus and $s \in P_0$ means the sentence came from the baseline (trigram language model here). A new function $h(s) = P(Y = 1|s)$ is defined and by Bayes theorem

$$h(s) = P(Y = 1|s) = \frac{P(s|Y = 1)P(Y = 1)}{P(s|Y = 1)P(Y = 1) + P(s|Y = 0)P(Y = 0)} \tag{2.26}$$

Assuming uniform distribution for prior probability of Y,

$$\frac{h(s)}{1 - h(s)} = \frac{P(s)}{P_0(s)} \tag{2.27}$$

using the exponential form of $P(s)$ in Equation (2.23) and taking logarithms gives

$$\begin{aligned} \log(\frac{h(s)}{1-h(s)}) &= -\log Z + \sum_i \lambda_i f_i(s) \\ &= \beta_0 + \sum_i \beta_i f_i(s). \end{aligned} \tag{2.28}$$

The Equation (2.28) has the form of logistic regression, when $f_i(s) = x_i$ and the sentence probability can be written as

$$\hat{P}(s) = P_0(s) exp(\hat{\beta}_0 + \sum_i \hat{\beta}_i x_i) \tag{2.29}$$

In information retrieval, the user's information need and preferences can be incorporated in the query model [37]. The complex aspects of information need and preferences can be viewed as diverse features that should be represented in the query model. The whole sentence maximum entropy model is the motivation for the whole document maximum entropy models proposed in Chapter 5. A variety of features, not necessarily independent, can be incorporated in the query or relevance model and the document-likelihood probability is estimated as a maximum entropy langugage model.

Maximum entropy models have been explored for the text classification problem. The next section briefly identifies the prior work in text classification using statistical language models and maximum entropy models.

2.4 Language Models for Text Classification

Language modeling approaches have been explored in the related problem of text classification. The problem of text classification is to categorize a document into one (or more) predefined classes in the set $C = \{c_1, c_2, \ldots, c_k\}$. Labeled document collections identifying documents belonging to the classes in set C are used in learning the classifiers for the classes. A "bag of words" representation of documents is used in text categorization and the words or features are used to determine the membership of a document to a class. In binary text classification the task is to identify if a document is in a particular class or not. This is similar to the problem of determining if a document is relevant to a given query. A number of classification techniques including linear classifiers and support vector machines have been explored for text classification. Naive Bayes classification is a most popular method for text classification. In estimating the conditional probability of a class label given a document, the Bayes rule is applied to generate

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}. \tag{2.30}$$

Representing the document d as a vector of terms $\{t_1, t_2, \ldots, t_n\}$, the Naive Bayes classifier makes the term independence assumption, given the class label, c and assigns the class to the document based on the maximum posterior probability

$$c^* = \arg\max_{c \in C} P(c|d) = P(c) \prod_{j=1}^{n} P(t_j|c) \tag{2.31}$$

If the prior probability for the class label is assumed to be uniform, then Equation (2.31) corresponds to a maximum likelihood naive Bayes classifier. Several variants of the naive Bayes classifier have been experimented with including binary independence model, multinomial model and Poisson model. The multinomial model seems to be the best choice [44] for text classification. It estimates the probability of a document given its class by

$$P(d|c) = |d|! \prod_{j=1}^{n} \frac{P(t|c)^{n(t,d)}}{n(t,d)!} \tag{2.32}$$

where $P(t|c)$ is the term probability in the class, c, and $n(t,d)$ is the frequency of term, t, in document, d. This model is the unigram language model in statistical language modeling. N-gram language models have been explored to extend the Naive Bayes classifier [54].

Recently, maximum entropy models have been applied to text categorization problems [52]. the conditional class probability of a document is estimated as a maximum entropy language model satisfying some constraints on real-valued feature functions defined for document-class pair. The labeled training data is used to estimated the expected values of the feature functions. The conditional distribution $P(c|d)$ estimated using MaxEnt should satisfy

$$\frac{1}{|D|}\sum_{d \in D} f_i(d, c(d)) = \sum_{d} P(d) \sum_{c} P(c|d) f_i(d, c) \tag{2.33}$$

The prior document probability is assumed to be uniform and the features are defined for each word-class pair as

$$f_{w,c'}(d, c) = \begin{cases} 0 & \text{if } c \neq c' \\ \frac{n(w,d)}{|d|} & \text{otherwise.} \end{cases} \tag{2.34}$$

Nigam et al [52] report the classification results of the MaxEnt model on three datasets and compare it with the Naive Bayes model. They report improvement in one of the datasets using MaxEnt and with the use of exponential prior for the feature weights to handle overfitting and feature selection problems, report maximum entropy to perform better than Naive Bayes in two of the three datasets. Similar to the whole sentence language models (Section 2.3.2), this model for text classification can support any real-valued feature function defined on the document-class pairs.

This chapter presented prior work in information retrieval and related topics in speech recognition, concentrating on the statistical language modeling approaches. Among language modeling approaches to information retrieval, the query-likelihood and document-likelihood approaches are of interest in this dissertation. Smoothed unigram language models are simple query-likelihood models for information retrieval. Chapter 3 explores models beyond unigram features and recognizes the need to identify *IR-specific* features. Language models that worked well for speech recognition were designed to satisfy the particular requirements of the application. In the case of IR, language models have to be *designed* to incorporate information retrieval specific requirements. Some aspects of statistical phrases in user query is used in the proposed query-likelihood models.

Moving from statistical to syntactic phrases, Chapter 4 proposes a two-layer query-likelihood model for information retrieval. Syntactic phrases are identified as "concepts" of interest to the user and the particular surface form used in the query is a particular realization of the concept that can occur in a relevant document.

A number of other textual and non-textual factors/features influence the relevance of a document to a given query. Chapter 5 propose a new model based on maximum entropy principle to incorporating such diverse features in a retrieval model. It is an attempt to move towards more semantic models for information retrieval as the proposed model enables statistical and semantic information about user's information need and documents be represented in a unified framework for information retrieval.

Chapter 3

Exploiting Term Co-occurrence for IR

Term occurrence statistics have been extensively used in different information retrieval models, including language modeling approaches. These simple models are easy to implement and in most cases, provide appreciable retrieval performance. However, the use of additional information beyond term occurrences have resulted in more precise language models for speech recognition. While the language modeling requirements of speech recognition are different from that of information retrieval, this chapter proposes methods for addressing some of these differences and "design" language models for information retrieval. N-gram models, that have been very effective in speech recognition, have been explored for information retrieval. Similar to N-gram language models, the biterm language models (Section 3.1) and the maximum bigram language models (Section 3.2) are language models based on term co-occurrence statistics. They capture different aspects of term co-occurrence that seem to be relevant to the particular requirements of information retrieval.

3.1 Biterm Language Models

The language modeling approach associates a language model for each document and views querying as a generative process. Given a query, $q = q_1, q_2, \ldots, q_k$, a document, d, is ranked based on the probability, $P(q|M_d)$ of its model, M_d, generating the given query. Smoothed unigram language models discussed in Section 2.2.2.1 make term independence assumption and compute a smoothed probability estimate for the term probability, $P(q_i|M_d)$.

Language models have been explored to incorporate more contextual evidence in query-likelihood models. N-gram language models, that have shown significant performance improvement in speech recognition was the first natural choice. In the context of query-likelihood models, the query-likelihood probability using a bigram language model for documents is given by

$$P(q|M_d) = \prod_{i=1}^{k} P(q_i|q_{i-1}, M_d) \tag{3.1}$$

In N-gram language models the term order or sequence is important. A smoothed bigram model using linear interpolation is given by

$$P(q_i|q_{i-1}, M_d) = \lambda_1 \ \hat{P}_2(q_i|q_{i-1}, d) + (1 - \lambda_1) \ \hat{P}_1(q_i|d) \tag{3.2}$$

Miller et al. [46] use the maximum likelihood estimate for $\hat{P}_2$ and a linear interpolation estimate for $\hat{P}_1$. They also motivate their solution using a 3-state Hidden Markov Model corresponding to “Document unigram”, “Document bigram” and “General English”. The transition probabilities are given by the parameters $\{\lambda_i\}$ and the output distribution are given by the corresponding maximum likelihood estimates. The maximum likelihood estimate for bigram (q_{i-1}, q_i) is given by

$$\hat{P}_{ml}(q_i|q_{i-1}, d) = \frac{c(q_{i-1}, q_i, d)}{c(q_{i-1}, d)} \tag{3.3}$$

where $c(q_{i-1}, q_i, d)$ is the number of occurrence of the bigram (q_{i-1}, q_i) and $c(q_{i-1}, d)$ is the occurrence count for the term q_{i-1} in document d. A similar model was proposed by Song and Croft [69] who use Good-Turing smoothing for estimating the probabilities.

Limited success was achieved in improving the retrieval performance using the smoothed N-gram language models. One possible reason for the absence of significant improvements using the additional information is that the models need to be "designed" for the application at hand.

3.1.1 Word Order Constraint

Smoothed N-gram language models give importance to the order in which words appear in documents. The order constraint seems relevant to applications like speech recognition where one is interested in predicting the next word based on the current and previously seen words. Such an assumption for language models for information retrieval is restrictive since the order in which words appear in user queries need not necessarily match the order in which they may appear in relevant documents. While query terms that occur together can be assumed to form bigrams for query representation, they may not appear together in the document in that order. For example *Information Retrieval* and *Retrieval of Information* refer to the same idea or concept, however, insisting on the order of occurrence (assuming stop words like *of* are removed) provides two different representation for the same concept. While all these surface forms infer the same concept of information retrieval, their likelihood estimates will be different using the bigram model. In biterm language models proposed here a document containing *information retrieval* and another one containing *retrieval of information* will be assigned the same probability of generating the query.

To distinguish unordered word-pairs from ordered word-pairs or *bigrams* in statistical language modeling terminology, we refer to the former as *biterms*. Unordered word-pairs have been explored as document features for document retrieval [47] in vector space models as well as text categorization [11] applications. Mitra et al. [47] explored the use of statistical and syntactic phrases as indexing terms in a vector space retrieval model. Each new phrase is identified by a particular dimension in the vector space and they are weighted based on their occurrence counts. Instead of using a multi-word phrase as an index term, it was represented by a set of unordered word-pairs. For example, *Computer Science Department* was represented by index terms *computer science* and *department science*. Mitra et al. found the phrases to be good indexing terms; However, they obtained mixed results in terms of retrieval performance. At high ranks, the number of single term matches are high resulting in substantial overlap of the subject matter of query and document. Including Phrase match in this setting over-emphasizes a particular match and retrieval to reflect the same. While the effect of phrases can be controlled using a phrasal weight in their weighting scheme, adhoc selection leads to less robust estimates for the contribution of the phrase to the similarity of document to given query. Mitra et al. used syntactic and statistical phrases in their experiments. Statistical phrases are used in biterm language models for representing the queries with robust probability estimation method to compute the query-likelihood probabilities. Concept language models proposed in Chapter 4 use syntactic phrases in query representation.

3.1.2 Biterm Probability Estimation

Biterm Language Models for information retrieval are query-likelihood models with likelihood estimation function similar to that of bigram language models in (3.1). Three approximation methods are proposed for computing the biterm probability for a word pair $\{q_{i-1}, q_i\}$.

A word pair $\{q_{i-1}, q_i\}$ can occur in text as two different ordered pairs: (w_{i-1}, w_i) and (w_i, w_{i-1}). The first approximation for biterm probability is given by the average of the bigram probabilities (P_{bg}) of the ordered pairs.

$$P_{BT1}(w_{i-1}|w_i, d) \approx \frac{1}{2}\left[P_{bg}(w_{i-1}|w_i, d) + P_{bg}(w_i|w_{i-1}, d)\right] \tag{3.4}$$

This approximation is based on the assumption that an average of robust estimates of bigram probabilities of the ordered pairs would result in a robust estimate of biterm probability.

A natural method for estimating the likelihood estimate is by counting. The second approximation is based on computing biterm probabilities from the word-pair frequency in documents. It is similar to the bigram probability in (3.3). The biterm probability of term pair $\{w_{i-1}, w_i\}$ is computed as the ratio of the number of occurrences of the term pair $\{w_{i-1}, w_i\}$ to that of the occurrence count of the term w_{i-1} in document d.

$$P_{BT2}(w_i|w_{i-1}, d) \approx \frac{C(w_{i-1}, w_i|d) + C(w_i, w_{i-1}|d)}{2\, C(w_{i-1}|d)} \tag{3.5}$$

In P_{BT2} the denominator depends on the occurrence count of w_{i-1}. To make the approximation truly independent of the term order, it can be set to the ratio of the frequency of the term-pair to the minimum of the frequencies of terms w_{i-1} and w_i. The use of minimum frequency increases the effect of term-pairs on the probability of a term sequence in a document.

$$P_{BT3}(w_i|w_{i-1},d) \approx \frac{1}{2}\left(\frac{C(w_{i-1},w_i|d)+C(w_i,w_{i-1}|d)}{\min\{C(w_{i-1}|d),C(w_i|d)\}}\right) \tag{3.6}$$

P_{BT2} and P_{BT3} are *ad-hoc* approximations for biterm probabilities.

In the experiments using biterm language models, smoothed language models were estimated based on the likelihood estimates above. Linear interpolation smoothing is employed to smooth biterm probabilities using the smoothed unigram probabilities as in

$$P(w_i|w_{i-1},d) = \alpha_1 P_{BT}(w_i|w_{i-1},d) + (1-\alpha_1)P(w_i|d) \tag{3.7}$$

$$P(w_i|d) = \alpha_2 P_1(w_i|d) + (1-\alpha_2)P(w_i|C) \tag{3.8}$$

where α_1 and α_2 are the combination weights. While these weights can be learned from some training data of queries and relevant documents [46], fixed weights were used in our experiments. The weight values selected in the experiments are consistent with weighting selected in similar experiments by other researchers. In retrieval using unigram language models, smoothing based on Dirichlet priors performs consistently well on different document collections. Both linear interpolation and Dirichlet priors are used in the experiments for smoothing the unigram language model $P(w_i|d)$ in Equation (3.7).

3.1.3 Experimental Results and Observations

The experiments evaluate the effectiveness of biterms as features capturing term co-occurrence statistics and how the different approximation for the biterm probabilities fare on different document collections. The Text Analysis and Processing for Information Retrieval (TAPIR) toolkit was used in the experiments. TAPIR is a software library that includes a collection of programs for tokenizing/parsing text documents, indexing large document collections and perform retrieval and text classification experiments. IR experiments are conducted by identifying the different indexing and retrieval components that should be used and the parameters for configuring these components. In addition to vector space models, the TAPIR toolkit supports different language modeling approaches. All experiments presented in this dissertation were performed using the TAPIR toolkit.

Experiments were conducted on two standard document collections. The first collection is the Wall Street Journal (WSJ2) subset of TREC4 test collection that has been used in a number of IR experiments using language modeling approach. This collection has 74,520 documents with around 250MB of data. The TREC4 queries (201–250) were used in the evaluation. The second corpus is the complete TREC4 document collection from TREC disks 2 and 3. It has 567,529 documents with 2.5GB of data. The TREC4 queries (201–250) were used in the experiments. The document collection are indexed using the TAPIR toolkit with words stemmed using an implementation of porter stemmer and stop-words removed before the index is created.

Table 3.1 compares the retrieval performance using smoothed unigram, bigram and the different approximations of the biterm language models. SUG-JM is a smoothed unigram language model that combines a document model with corpus model through linear interpolation of term probabilities with the weighting factor of 40% set to the document model. BG-JM is a smoothed bigram language model which combines bigram document model and smoothed unigram language model. The weighting parameter between document and corpus models in the unigram model is set to 40% and the weighting parameter for bigram document model set to 10%.

Biterm retrieval systems were implemented with different approximations for biterm probabilities. The interpolation parameters were set at 40% for document model over corpus model and 10% for biterm models over unigram models. For smoothing using Dirichlet priors the value of μ was set to 1000. BT1-JM uses average of bigram probabilities as given by (3.4). BT2-JM is the biterm language model using the *ad-hoc* probability given by (3.5) and BT3-JM corresponds to the biterm probability in (3.6). These runs are smoothed using linear interpolation or Jelinek-Mercer smoothing. BT1-DIR, BT2-DIR and BT3-DIR are the corresponding runs where the unigram language are smoothed using Dirichlet priors.

Table 3.1 shows the comparison of non-interpolated average precision between different retrieval systems on the WSJ2 data set using the Jelinek-Mercer Smoothing. The biterm approximations provide improvements of 8 to 16% in average precision. BT3-JM improvement over smoothed unigram model is statistically significant using the Wilcoxon one-sided test at 0.05 level. The statistical improvement is indicated by the star mark in the corresponding entry. The average precision values are comparable to that of smoothed bigram language model. The BT2 approximation using occurrence counts performs better than other approximations. However, the difference between different biterm approximations and the bigram language model are not statistically significant.

Run	AvgP	% chg	Recall	% chg	RPrec	% chg	InitPr	% chg
SUG-JM	0.2176	-	951/1183	-	0.2267	-	0.5562	-
BG-JM	0.2524	15.99	948/1183	-0.32*	0.2711	19.59*	0.5956	7.08*
BT1-JM	0.2354	8.18	946/1183	-0.53*	0.2380	4.98	0.5884	5.79
BT2-JM	0.2540	16.73	942/1183	-0.95*	0.2628	15.92*	0.5862	5.39*
BT3-JM	0.2534	16.45*	951/1183	0.00*	0.2575	13.59*	0.6281	12.93*

Table 3.1: Experimental results for Biterm Language Models on the WSJ2 data set

Table 3.2 provides a comparison of different performance measures between the smoothed unigram language model and biterm language model using the BT3 approximation. The first two columns give the precision values at different recall levels and precision after N documents are retrieved for the two retrieval models. The third column gives the percent change in values. Column four gives the significance value according to the one-sided Wilcoxon test. The fifth column gives the number of queries for which the performance improved (I) and the number of queries for which the performance was different (D). The last column gives the significance values according to the one-sided Wilcoxon test on the queries for which the particular statistic for BT3-JM improved over SUG-JM. This column indicates whether the performance improvement is significant if we consider only those queries for which the performance of BT3-JM did improve over SUG. In columns four and six, a star mark after the significance value indicates a statistical significant difference at the 0.05 level.

The improvements indicated in the Table 3.2 are statistically significant in different performance metrics. An improvement of 16.5% in average precision is statistically significant at the 0.05 level in one-sided Wilcoxon test. Figure 3.1 illustrates the improvements using the biterm language models with approximations BT2 and BT3 over smoothed unigram language models.

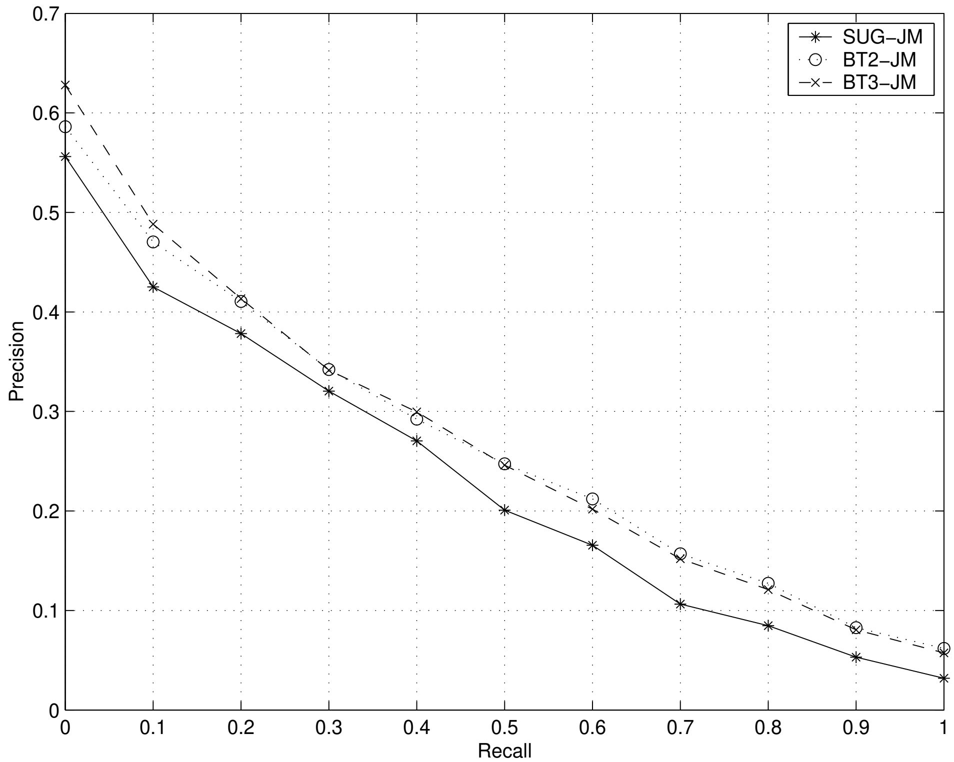

Figure 3.1: Precision-Recall Curve for WSJ2 collection comparing biterm and unigram language models smoothed by linear interpolation

Comparison of the different retrieval models using the Dirichlet smoothing is given in Table 3.3. Comparable performance improvements are observed using biterm language models over smoothed unigram models and small improvement over bigram models using the BT3 approximation. The improvements using the Dirichlet smoothing are statistically significant for some performance metrics as shown in Table 3.4. An increase of 9.6% in average precision is not statistically significant. However, the average precision value increased in 20 of the 45 queries.

	SUG-JM	BT3-JM	%chg	Wilc.	I/D	I-Wilc.
Rel	1183	1183				
Rret	951	951	0.0	0.0001*	5/12	0.0001*
0.00	0.5562	0.6281	12.9	0.0004*	15/24	0.0000*
0.10	0.4252	0.4883	14.8	0.0138*	18/35	0.0005*
0.20	0.3782	0.4134	9.3	0.0209*	20/36	0.0012*
0.30	0.3205	0.3415	6.6	0.0780	20/38	0.0122*
0.40	0.2704	0.2996	10.8	0.0816	19/40	0.0031*
0.50	0.2008	0.2461	22.6	0.0186*	20/39	0.0001*
0.60	0.1656	0.2018	21.9	0.0009*	21/37	0.0000*
0.70	0.1064	0.1519	42.8	0.0018*	18/35	0.0000*
0.80	0.0848	0.1210	42.7	0.0005*	14/30	0.0000*
0.90	0.0531	0.0806	51.8	0.0001*	11/21	0.0000*
1.00	0.0319	0.0574	79.9	0.0000*	8/13	0.0000*
Avg	0.2176	0.2534	16.5	0.0203*	25/43	0.0000*
5	0.3022	0.3289	8.8	0.0001*	9/17	0.0000*
10	0.2756	0.2956	7.3	0.0001*	12/20	0.0000*
15	0.2370	0.2681	13.1	0.0000*	16/23	0.0000*
20	0.2122	0.2411	13.6	0.0000*	20/28	0.0000*
30	0.1874	0.2081	11.0	0.0000*	18/28	0.0000*
100	0.1120	0.1158	3.4	0.0021*	20/32	0.0000*
200	0.0748	0.0762	1.9	0.0013*	16/27	0.0001*
500	0.0383	0.0379	-1.0	0.0192*	8/22	0.0083*
1000	0.0211	0.0211	0.0	0.0109*	5/12	0.0072*
RPr	0.2267	0.2575	13.6	0.0009*	17/25	0.0001*

Table 3.2: Comparison of Smoothed Unigram Language Model and Biterm Language Models using BT3 approximation for WSJ2 corpus. Models smoothed using Linear Interpolation.

The *adhoc* approximations perform better than bigram language models for the WSJ2 data set. The biterm and bigram contributions are set at 10%. Experimenting with different biterm-unigram combination weights (α_1 in Equation (3.7)) indicates that the best average precision is obtained at lower α_1 values. Fixing the smoothing of unigram language model using linear interpolation with corpus contribution at 60% and varying the biterm-unigram combination weights, at $\alpha_1 = 0.2$, the average precision using BT3 approximation is 0.2538. Higher values for α_1 reduces the average precision values (0.2495 for $\alpha_1 = 0.3$,

Run	AvgP	% chg	Recall	% chg	RPrec	% chg	InitPr	% chg
SUG-DIR	0.2282	-	973/1183	-	0.2395	-	0.6124	-
BG-DIR	0.2442	7.01	961/1183	-1.23	0.2475	3.34	0.6005	-1.94
BT1-DIR	0.2316	1.49	954/1183	-1.95	0.2233	-6.76	0.6002	-1.99
BT2-DIR	0.2426	6.31	961/1183	-1.23	0.2489	3.92	0.5671	-7.40
BT3-DIR	0.2502	9.64	964/1183	-0.92	0.2449	2.25	0.6046	-1.27

Table 3.3: Experimental results using Dirichlet smoothing for Biterm Language Models on the WSJ2 data set

0.2451 for $\alpha_1 = 0.4$, etc.) Term occurrence statistics provides lot of support for robust estimation of the biterm and bigram probabilities.

The reduction in average precision for biterm language models using BT1 approximation in comparison to bigram language models can be attributed to the averaging of probabilities reducing the effect of term pairs. The *ad-hoc* approximation P_{BT2} for biterms improves slightly over bigram language model since it ignores the order of term-occurrence while computing the frequency of term-pair occurrence. If w_{i-1} occurs more often than w_i in the term pair $\{w_{i-1}, w_i\}$ in a document, the effect of the co-occurrence of term-pair to the biterm probability is reduced. Choosing the minimum of the term occurrence count for the denominator as in BT3 approximation has improved retrieval performance over the bigram language model.

TREC4 is a larger corpus than WSJ2. The improvements obtained using bigram and biterm language models are not appreciable with either of the smoothing methods. Table 3.5 gives average precision values for different language models using linear-interpolation and Table 3.6 corresponds to smoothing using Dirichlet priors. Some of the biterm approximation do not perform as well as smoothed unigram language models.

	SUG-DIR	BT3-DIR	%chg	Wilc.	I/D	I-Wilc.
Rel	1183	1183				
Rret	973	964	-0.9	0.0865	6/15	0.0576
0.00	0.6124	0.6046	-1.3	0.4158	9/23	0.2826
0.10	0.4667	0.4832	3.5	0.1857	15/36	0.0644
0.20	0.3890	0.4251	9.3	0.1276	16/36	0.0321*
0.30	0.3231	0.3352	3.7	0.4354	18/38	0.1556
0.40	0.2758	0.2992	8.5	0.3773	18/41	0.0884
0.50	0.2024	0.2475	22.3	0.0295*	22/41	0.0003*
0.60	0.1691	0.1948	15.2	0.3218	15/39	0.0469*
0.70	0.1119	0.1517	35.6	0.1079	15/35	0.0064*
0.80	0.0837	0.1234	47.4	0.0111*	15/31	0.0004*
0.90	0.0539	0.0779	44.5	0.0060*	9/22	0.0015*
1.00	0.0365	0.0570	56.2	0.0036*	6/16	0.0019*
Avg	0.2282	0.2502	9.6	0.3584	20/42	0.0134*
5	0.3244	0.3689	13.7	0.0008*	13/18	0.0003*
10	0.2867	0.3133	9.3	0.0063*	15/24	0.0022*
15	0.2607	0.2830	8.6	0.0105*	17/31	0.0006*
20	0.2367	0.2467	4.2	0.0199*	13/27	0.0036*
30	0.1985	0.2141	7.9	0.0062*	17/30	0.0007*
100	0.1162	0.1156	-0.5	0.0684	15/31	0.0083*
200	0.0783	0.0769	-1.8	0.0494*	14/27	0.0211*
500	0.0387	0.0384	-0.8	0.1022	11/23	0.0527
1000	0.0216	0.0214	-0.9	0.0779	6/15	0.0569
RPr	0.2395	0.2449	2.3	0.2721	13/25	0.1090

Table 3.4: Comparison of Smoothed Unigram Language Model and Biterm Language Models using BT3 approximation for WSJ2 corpus. Models smoothed using Dirichlet priors

Comparing the average precision performance measure on a query-by-query basis, Figure 3.2 illustrates that biterm and bigram perform better that unigram models in a number of queries. For some queries, the performance reduces drastically (for example, Topic 214: *What are the different techniques used to create self-induced hypnosis?*). The similarity of variation in the difference values in average precision of biterm and bigram over smoothed unigram language models in Figures 3.2 (a) and 3.2 (b), respectively, indicate the effect of term co-occurrence on retrieval. Figures 3.2 (c) shows the difference

Method	AvgP	%Change	%Change
SUG-JM	0.1902	-	-
BG-JM	0.1953	2.68	-
BT1-JM	0.1828	-3.89	-6.40
BT2-JM	0.1845	-3.00	-5.53
BT3-JM	0.1955	2.79	0.10

Table 3.5: Experimental results using Jelinek-Mercer Smoothing for Biterm Language Models on the TREC4 data set

Method	AvgP	%Change	%Change
SUG-DIR	0.1876	-	-
BG-DIR	0.1938	3.30	-
BT1-DIR	0.1790	-4.58	-7.64
BT2-DIR	0.1915	2.08	-1.19
BT3-DIR	0.1929	2.83	-0.46

Table 3.6: Experimental results using Dirichlet smoothing for Biterm Language Models on the TREC4 data set

between the average precision values of BT3 approximation and bigram language model that are not significant.

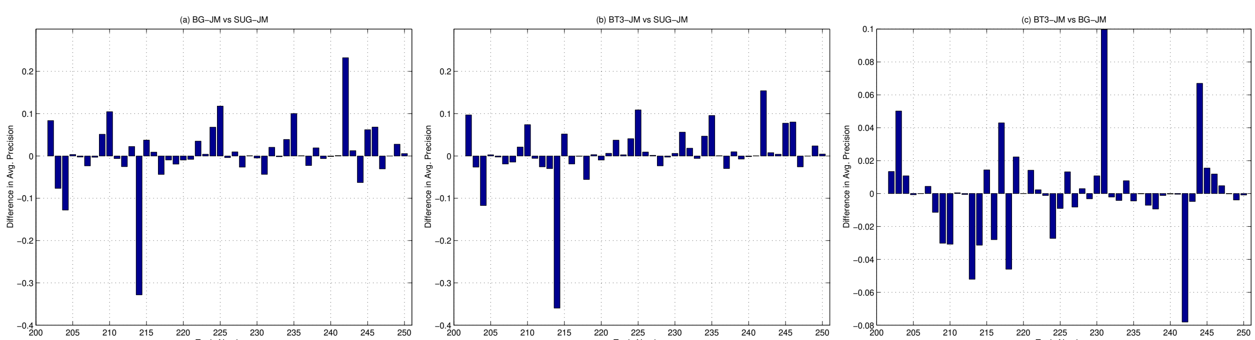

Figure 3.2: Difference in Average Precision Plots for TREC4 collection comparing biterm and bigram and unigram language models smoothed by linear interpolation

Biterm language models capture term co-occurrence and improve retrieval performance over smoothed unigram and bigram language models. While the maximum likelihood estimate of a bigram is naturally defined, approximate measures have to be defined to estimate the biterm probability based on occurrence counts. The word order constraint is an important aspect for incorporating term co-occurrence information for retrieval. Relaxing this constraint addresses the surface form variability in queries. If a particular concept can be expressed in more than one surface form, as in *computer science department* and *department of computer science*, biterm language models are able to capture them correctly. While ignoring the word order in word-pairs captures the user's intent is most cases, for some phrases their meaning changes with the change in the word order. For example, a *blind venetian* is not a *venetian blind*. Maximum bigram language models, discussed next, use collection statistics to handle such term dependencies.

3.2 Maximum Bigram Language Models

N-gram, specifically, bigram language models that capture context provided by the previous word(s) perform better than unigram models [69]. Biterm language models (Section 3.1 that ignore the word order constraint in bigram language models have been shown to perform slightly better than bigram models. However, word order constraint cannot always be relaxed for all phrases since a *race horse* is different from an *horse race* and a *victim of juvenile crime* not necessarily a *juvenile victim of crime*. A shallow parse of the user query can identify part-of-speech of the query words and group them into syntactic and possibly, semantic groups of words. This is exploited in Concept Language Models discussed in Chapter 4. Here, the maximum bigram language models exploit co-occurrence statistics to model term dependencies for information retrieval.

Early work on using term dependencies in information retrieval is due to van Rijsbergen [79]. In the context of probabilistic retrieval models, term dependencies were introduced in binary independence models by representing sentences in a document as a maximum spanning tree (MST). The words or terms appearing in a document are viewed as nodes of a graph and edges connecting them are weighted based on the dependence or cohesion between two nodes or terms. Van Rijsbergen used the *Expected Mutual Information Measure* (EMIM) to measure term dependencies. The MST of the graph identifies the dependencies between terms in a sentence. A similar method was employed in building word-trigger based language models for speech recognition applications [62] that incorporate long distance term dependencies beyond those captured by an N-gram model. Term dependencies in each sentence of a document in the collection is captured using the maximum spanning tree representation. Such representations were used in retrieval experiments on small document collections and significant improvements over independence retrieval models were reported. Larger collection would require more indexing time to generate the MST for the sentences in all the documents in the collection.

Recently, Nallapati and Allan [48] proposed Sentence Tree based Language Model that used the MST representation for a sentence in a document for the Story Link Detection task in TDT. Given two randomly selected documents, the story link detection task is determine if they discuss the same topic. A document is viewed as a collection of sentences. By defining a topic model for a document, document similarity is measured by the likelihood of generating sentences of the second document using the topic model. Sentence likelihood is measured by a bigram approximation with term dependencies learned by representing the sentence as an MST. The dependencies between terms in a sentence is measured using the Jaccard coefficient

$$J(w_i, w_j) = \frac{n(w_i, w_j)}{n(w_i) + n(w_j) - n(w_i, w_j)} \tag{3.9}$$

where $n(w_i)$ is the number of sentences in which w_i occurs and $n(w_i, w_j)$ is the number of sentences where w_i and w_j co-occur. The Jaccard coefficient is used as a measure for query term dependencies in the following.

3.2.1 Query Term Dependencies

The bigram and biterm language models capture local dependencies between query terms, specifically with the previous and next word in the query. In keyword queries, typical of web search, such dependencies may not reflect the user's information need. While the TREC queries used in the experiments are well-formed, the co-occurrence statistics learned from documents may not match the term dependencies in the query.

In query likelihood model, the probability of a document, d, given the query, $Q = q_1 q_2 \cdots q_k$, is

$$P(Q|M_d) \approx \prod_i P(q_i|q_1, q_2, \ldots, q_{i-1}, M_d). \tag{3.10}$$

In bigram language models, the query-term probability given the history or previous words is approximated to depend only on the previous word. This strict dependence on the previous word is relaxed in maximum bigram language models by setting

$$P(q_i|q_1, q_2, \ldots, q_{i-1}, M_d) \approx \max_{j=1,\cdots,i-1} P(q_i|q_j, M_d) \tag{3.11}$$

where the q_j corresponding to the maximum probability value can be viewed as the *best* trigger term for predicting q_i. Defining complete dependence of a query term, q_i, on previous words, $q_1, q_2, \cdots, q_{i-1}$, the most pronounced dependence in the document model between a query term q_j in the history and q_i is selected and used in the estimation.

Note that the selection of the *best* trigger for q_i depends on the document model. The first query term is arbitrarily selected to the independent term in this representation. Such a selection is true for bigram language models also. A better selection criteria for the independent term or the root of the spanning tree is left for further investigation. In the experiments, the maximum bigram language model is compared with a query-likelihood model where the term dependencies are learned using the Jaccard coefficient in (3.9).

3.2.2 Experimental Results and Observations

The experiments on TREC collections evaluate the effectiveness of the maximum bigram language models and compares them to the query likelihood model that incorporates term dependencies using Jaccard coefficient. The experiments are performed on the Wall Street Journal (WSJ2) subset of TREC4 test collection and the TREC4 test collection. TREC4 topic queries were used in the evaluation.

Table 3.7 summarizes average precision and R-precision values for different retrieval methods on the WSJ2 dataset with language models smoothed using Dirichlet priors. Smoothing weights were manually selected similar to the weights used in biterm language model experiments. The Dirichlet prior value was set to 1000 and the bigram contribution was set at 10%. MBG-DIR corresponds to the maximum bigram language model given in equations (3.10) and (3.11). MBGJ-DIR uses the maximum spanning tree induced on the query using the Jaccard coefficient (3.9).

Method	AvgP	%Change	%Change	RPres	%Change	%Change
SUG-DIR	0.2282	-	-	0.2395	-	-
BG-DIR	0.2442	7.01	-	0.2475	3.34	-
MBG-DIR	0.2523	10.56	3.32	0.2716	13.40	9.74
MBGJ-DIR	0.2221	-2.67	-9.05	0.2415	0.84	-2.42

Table 3.7: Comparison of average precision and R-precision values for unigram, bigram and maximum bigram models smoothed using Dirichlet priors for WSJ2 dataset

Figure 3.3 illustrates the precision-recall curves for the four different retrieval methods. The maximum bigram language model gives almost 11% improvement in the average precision values over smoothed unigram language models. The improvement is not significant over bigram language models. The maximum bigram language models are a slight modification of the bigram language models with the concept of best triggers rather than using the previous word as the term the current term depends on.

The maximum bigram language models do not perform as well for TREC4 document collection. Table 3.8 compares the average precision and R-precision values for the different retrieval methods using Dirichlet smoothing.

Method	AvgP	%Change	%Change	RPres	%Change	%Change
SUG-DIR	0.1876	-	-	0.2475	-	-
BG-DIR	0.1938	3.30	-	0.2516	1.66	-
MBG-DIR	0.1848	-1.49	-4.64	0.2419	-2.26	-3.86
MBGJ-DIR	0.1836	-2.13	-5.26	0.2458	-0.69	-2.31

Table 3.8: Comparison of average precision and R-precision values for unigram, bigram and maximum bigram models smoothed using Dirichlet priors for TREC4 corpus

A comparison of difference in the average precision values is given in Figure 3.4.

Selecting the best trigger words for a query term is important. For a larger collection, the simple heuristic of using the previous word seems to work very well. Well-formed

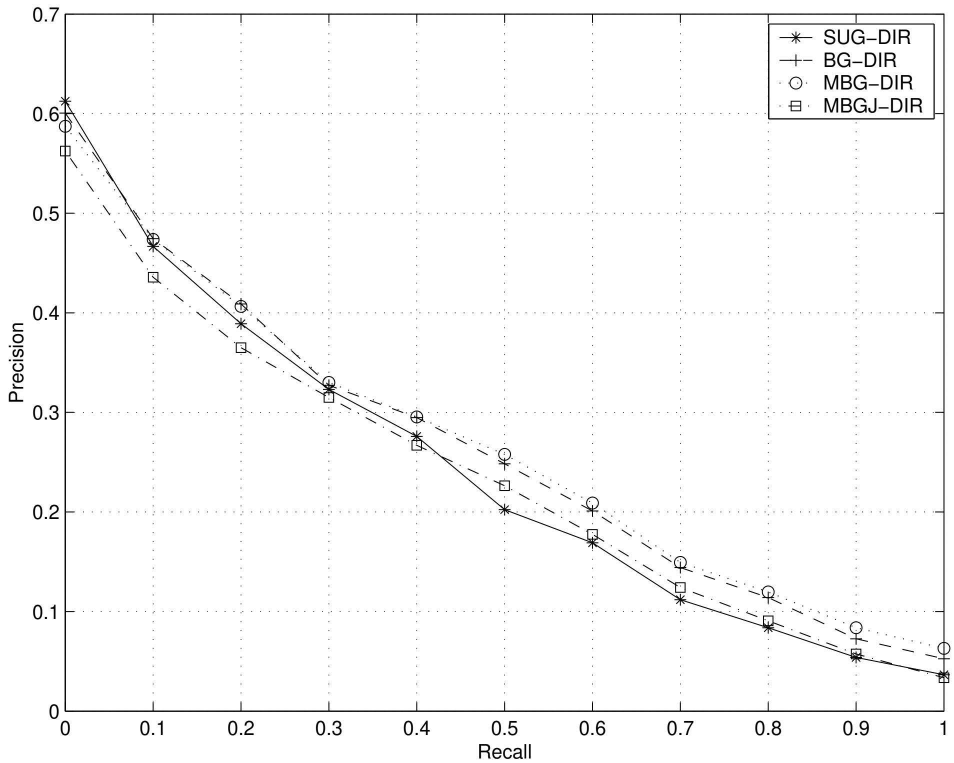

Figure 3.3: Precision-Recall curves comparing smoothed bigram and maximum bigram language models for WSJ2 collection

queries being used in the experiments contributes to this performance. However, incorporating term dependencies does help in improving retrieval performance.

3.3 Summary

This chapter introduced two language models for information retrieval that incorporate term co-occurrence information. N-gram models have been used in prior retrieval experiments based on language modeling approaches. The word order constraints that characterizes N-gram language models can be relaxed in language models for information retrieval. The biterm language models with three different approximation methods for estimating their probabilities were shown to provide some performance improvements. The

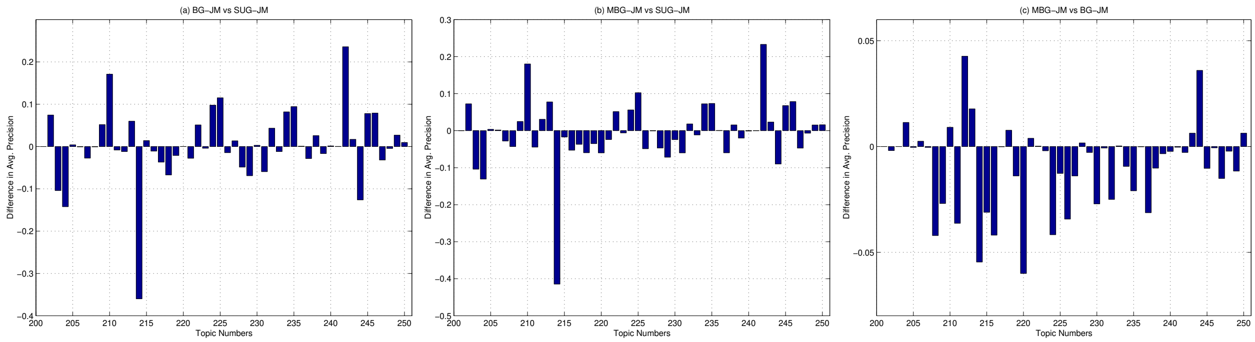

Figure 3.4: Difference in Average Precision Plots for TREC4 collection comparing maximum bigram and bigram and unigram language models smoothed by Dirichlet priors

word order constraint, however, cannot be relaxed arbitrarily as some phrase have different meaning when the order of the words in them are changed.

As an attempt to address the problem of detecting term dependencies among query terms, the maximum bigram language model lets the document model determine the dependencies by identifying the best trigger for a query term among the other words in the query. While they perform better than the term dependencies learned using the Jaccard coefficient, bigram language models where a query term is assumed to depend on the previous word perform better in larger collections. Capturing term dependencies in sentences, used in [48, 79], have sufficient evidence in the document corpus, better methods for learning term dependencies in queries needs to be investigated.

Retrieval models that can identify when to relax the word order constraint and when to enforce the constraint would result in improved retrieval performance. There has been some effort in identifying lexical atoms – phrases in which one or more constituent words do not carry their regular meanings (e.g. *hot dog*) [84]. Such phrases are discovered by exploiting the context in which the phrase and the constituent words appear. The context in which word pairs appear in different orders can be compared to decide on enforcing the word order constraint resulting in a combined bigram-biterm model for retrieval.

Biterm and maximum bigram language models are based on statistical phrases, specifically, words pairs, identified in the query text. Syntactic parsing of the queries would result in better representation based on phrases. The concept language models presented in the next chapter provide a new view of the query generation process that captures some of the query term dependencies in the retrieval model as derived from the syntactic phrases in the query.

Chapter 4

Concept Language Models

Term independence assumption, which is understood to be a wrong assumption for text retrieval, is used in most retrieval models for robust estimation and ease of computation. The biterm language models and the maximum bigram language models presented earlier in this chapter relax these assumptions. These and N-gram language models, to some extent, do capture term dependencies as defined by the proximity of term occurrence within the window of size N. However such local dependencies derived through co-occurrence statistics do not necessarily capture user's information need precisely. For example, in the query: *How often were the peace talks in Ireland delayed or disrupted as a result of acts of violence* (TREC8 topic 404), a document with the sentence *Sir Paul talks in Ireland about his new book...* will be deemed relevant for this query.

Assuming the model that a user generates the query text by selecting terms from what is perceived as a relevant document, the query term selection follows the model of this "ideal" relevant document. The ideal model would exhibit a number of characteristics, some captured best by, say, a unigram language model, and some more by a bigram model, and so on. The task, then, is to (1) identify these characteristics of the "ideal" model and (2) rank documents based on their model exhibiting such characteristics. The concept language model, presented in this section, use NLP techniques to identify characterizing candidate features, referred to as concepts, of the "ideal" relevant document from the user query and uses a combination of N-gram models, based on the identified concepts, to estimate the query-likelihood probabilities for documents.

The concept language model represents an alternate view of the query generation process. A query is viewed as an expression of different *concepts* and their relationships that are of interest to the user. The task of the retrieval system is to identify these *concepts* expressed in the user query and model their occurrence in documents for retrieval purposes. Retrieval performance depends on correctly identifying these concepts in queries and in documents. Concepts are not necessarily expressed by contiguous words in a query and in most cases, the *context* in which a concept appears is essential to correctly identifying the user's intent. For TREC topic 404, the context of the noun phrase, *peace talks*, is *Ireland* and this context should be expressed in a document to be relevant to the query. While sophisticated representation is required to precisely capture this implication, the concepts are assumed to be phrases identified by a syntactic parse of the query. Identifying the concepts in documents and creating a concept index is computationally expensive and hence the approach adopted here is to process the user queries only to identify concepts. Proximity constraints are used to identify the concept occurrences in documents.

4.1 NLP and IR

This section presents a brief review of prior work on using NLP techniques in IR (see [28] for a comprehensive survey). While NLP is implicitly used in stemming and generation of stop-word lists for IR, its use in identifying phrases either in documents and/or queries is of interest here.

The use of phrases as index terms has been explored extensively by the participants of Text REtrieval Conferences (TREC) [50]. Phrases seem, intuitively, to be good descriptors of document content. Both statistical (identified based on word co-occurrence) and syntactic (using some NLP techniques) methods have been explored to identify indexing phrases and experiments have shown them to be useful indexing units. Mitra *et al.* [47] compared statistical and syntactic indexing phrases and found them to be useful for certain queries and observed a trade off between query accuracy and query coverage. They also show that longer phrases do not give significant improvement in retrieval performance. Their justification is that a longer phrase match invariably includes the match of the two-word phrases extracted from it and overemphasizing a single longer match may hurt the performance of the system. Strzalkowski *et al.* [78] use multiple indices of documents based on stemmed words, phrases, named entities and headword-modifier pair and merge retrieval results on these different indices to generate a final ranked list of documents for a given query. Sparck-Jones [70] note that simple linguistically motivated indexing techniques turned out to be more effective than well-executed statistical approaches, at least for English, while more advanced NLP techniques such as concept extraction, remained too expensive for large-scale applications. While the output of the information extraction system used for the experiments does include named entities, events, etc., only the parse tree of the query is used to build concept language models.

Instead of processing documents through an NLP system to identify phrases for indexing, which is computationally expensive, there have been efforts to use linguistic processing of queries to get better representation and understanding of user's information need. Early efforts in this direction include the experiments by Smeaton and van Rijsbergen [68] by implementing a retrieval strategy that is based on syntactic analysis of queries. A syntactic parse of the query is used to identify dependent word pairs and the retrieval strategy is to search for co-occurrence of word pairs within a sentence in documents. Recently, Narita and Ogawa [49] examined the utility of phrases, extracted from query texts using NLP, as search terms for IR. They used single term and phrasal term in their query construction. Similar to Mitra *et al.* [47] they experimented with different representations for multi-word phrases (more than 2 words) and decided to use two word phrases for query construction. Phrasal terms were matched using proximity constraints. They also associated lesser weight to phrasal terms than single terms since occurrence of a phrase in a document also indicates the occurrence of its constituent words. While their decision to represent multi-word phrases as a set of two-word phrases is based on experimental results, one possible explanation for using such an approximation is the absence of robust estimate for term weight that is based only on the phrase occurrence in documents. A language model that can incorporate evidence from the document and corpus may be able to provide a more robust estimate for multi-word phrases. This work is the closest, in spirit, to the concept language models.

The above experiments on the use of phrases and other syntactic structures in IR either at indexing and/or retrieval time have been in the settings of vector space model. Each phrase and its constituent words are assigned a unique entry in the weighted vector for a document. While these entries or dimensions in the document vector are related, their dependencies are either ignored in the formulation or taken care of in an *ad-hoc* fashion by weighting the constituent words more than the phrases. Latent Semantic Indexing (LSI) [21] has been used for dimensionality reduction and capturing term dependencies. However, it is difficult to appreciate and understand the retrieval model underlying its representation. Language Modeling approach to IR, which attempts to explain the retrieval process seems to be a good framework to capture term dependencies. The concept language models is a new method for incorporating term dependencies discovered through syntactic parse of user query.

NLP techniques have been used for improving language models in speech recognition. Section 2.3.1 presented prior work in using syntactic and semantic information in generating language models for speech recognition. In the context of language modeling approach to IR, the methods to exploit NLP, beyond stemming and stop-word lists, of documents/queries for improving retrieval is in its early stage. Allan and Raghavan [2] use possible Part-of-Speech tags for words in a query for the purpose of disambiguating the query words and reduce query ambiguity. Given a user query, their solution is to generate a set of questions that will help the user to disambiguate the query and retrieve documents based on the disambiguated query. This requires the document collection to be parsed by an Part-of-Speech tagger and generate search index based on the document content and tags.

4.2 Query as a Sequence of Concepts

The query-likelihood language modeling approach to IR views a user query, Q, as a set or sequence of terms. A user query is an expression of user's information need. It is not the words themselves but the concepts that they refer to which are of interest to the user. A query, $Q = \{q_1, q_2, \ldots, q_n\}$, can be viewed as a sequence of concepts, $\{c_1, c_2, \ldots, c_k\}$, of interest to the user. While concepts may be expressed in a query through one or more words that are not necessarily occurring contiguously in the query, as a simplifying assumption, a concept, c_j, is viewed as query term sub-sequence, $\{q_1^j, q_2^j, \ldots, q_{n_j}^j\}$. As an example, the concepts in *Status of nuclear proliferation treaties — violations and monitoring.* (TREC4 topic 202) are *Status*, *nuclear proliferation <u>treaties</u>*, *violations* and *monitoring* with the headword of multi-word concepts underlined. Then, the query-likelihood probability, $P(Q|M_D)$, is given by

$$P(Q|M_D) = \prod_{j=1}^{k} P(c_j|c_1, c_2, \ldots, c_{j-1}, M_D). \tag{4.1}$$

The conditional concept probability in the right-side of (4.1) can be approximated to unigram model by assuming concept independence. This approximation, referred to as the *Concept Unigram Language Model* (CULM) is given by

$$P(Q|M_D) = \prod_{j=1}^{k} P(c_j|M_D) \tag{4.2}$$

Higher level models based on concept N-grams can be defined as approximations to (4.1) which take the concept context into account.

The concept unigram probability, expressed by the joint probability of query terms in the concept, is given by

$$P(c_j|M_D) = \prod_{l=1}^{n_j} P(q_l^j|q_1^j, q_2^j, \ldots, q_{l-1}^j, M_D) \tag{4.3}$$

If the right-side of (4.3) is approximated to product of query term unigram probability, $P(q_l^j|M_D)$, the resultant distribution is the unigram language model (2.6). The number of terms in a concept is query dependent. While a concept may have more than 2 terms, there may not be sufficient statistics to obtain a robust estimate of the joint probability in (4.3). Since the documents are not pre-processed by the NLP components to identify concept occurrence, alternate methods are used for counting concept occurrence and to estimate concept probability. The order in which the terms in a concept appear in a relevant document is likely to be different from their order in the query. For computational purposes and to incorporate dependency constraint between terms in a concept in the probability estimation, a bigram approximation of concept probability given by

$$P(c_j|M_D) \approx \prod_{l=1}^{n_j} P(q_l^j|q_{l-1}^j, M_D). \tag{4.4}$$

is used.

Concept language models is indeed a family of language models for information retrieval. Different models can be derived for estimating query-likelihood probabilities based on different representation for the query as a sequence of concepts. In the case of concept unigram language models, independence of concepts is assumed and concept probabilities are estimated based on local term dependency models like N-grams. It is possible to relax the concept independence assumption to generate higher-order concept models.

Smoothing using Dirichlet priors is used to obtain a robust estimate of the bigram approximation to concept probability in (4.4). In Dirichlet smoothing, the probability of the term t in document D is assigned some prior probability based on its corpus probability weighted by the parameter μ. The model is given by

$$p_\mu(t|D) = \frac{c(t, D) + \mu p(t|C)}{\sum_w c(w, D) + \mu} \tag{4.5}$$

where $c(t, D)$ is the occurrence count of t in D and $p(t|C)$ is the corpus probability of t. Dirichlet smoothing has been shown to result in the best unigram language models for IR [86] tested against different test collections.

Bigram probabilities are smoothed using linear interpolation with smoothing parameter λ as in

$$P(t_i|t_{i-1}, M_D) = \lambda \frac{c(t_{i-1}, t_i, D)}{c(t_{i-1}, D)} + (1 - \lambda) P(t_i|M_D). \tag{4.6}$$

where $c(t_{i-1}, t_i, D)$ gives the occurrence count for the ordered term pair (t_{i-1}, t_i) in document D.

4.3 Indexing Schemes

This section briefly discusses the indexing schemes used to support the estimation of query-likelihood probabilities using the concept language model. It has been shown that unigram language models can be estimated using the settings of a vector space model since they only require hit statistics [30]. N-gram language models with $N > 1$ can either be estimated by maintaining an N-gram index created during index time or can be computed during query time. To facilitate different approximations to the concept probability in (4.3), we adopted the latter approach. This seems to be the best option for an experimental setup.

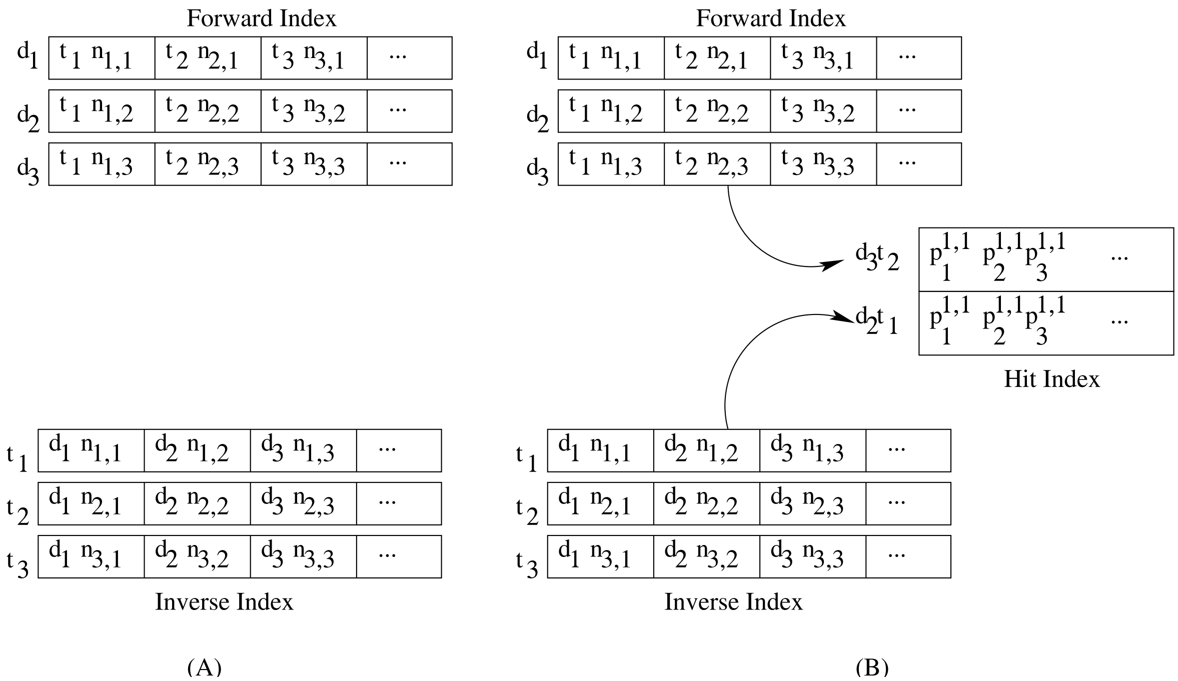

Figure 4.1: Index Structures - (A) Standard Forward and Inverse Indices; (B) Forward, Inverse and Hit Indices in TAPIR

Standard indexing schemes store for each term t a list of document indices in which the term appears. This is referred to as the inverted index. Figure 4.1(A) gives the standard indexing data structures of forward and inverse indices. The forward index stores the term and count information for each document and the inverse index stores the document and count information for each term. To enable query time computation of N-gram probabilities, a forward index is also maintained that stores a list of terms that appear in a document for each document. To compute N-gram statistics during query time, the positions at which a term appears in a document is required. This information is stored in a separate index, referred to as the *hit index*. This is illustrated in Figure 4.1(B). Given a document, d, and term, t, the hit index maintains the positions in which the term t appears in document d. Such an indexing scheme facilitates experimentation of different approximations to concept term probabilities. Query-time generation of document language models does take more time than using pre-indexed information. However, efficient indexing is needed to support higher order language models for document retrieval. These indexing data structures are supported in TAPIR toolkit.

4.4 Syntactic Parsing of Queries

The performance of the proposed Concept Language Models for IR depends on the effective parsing of the user queries. The parse tree of a user query is used to identify the concepts in the query. The restriction of concepts to contiguous words in the experiments results in concepts to be verbs and noun phrases in the query. While a part-of-speech tagger followed by shallow parsing of the POS output is sufficient to identify concepts, an information extraction engine, InfoXtract [71, 72], from Cymfony Inc. [1], was used to process the user queries. InfoXtract's output includes named entities, phrasal verbs, SVO structure, named associations, and events. Such rich information about the query is valuable for understanding the user's information need. However, only the shallow parsing results are used in the experiments.

[1] `http://www.cymfony.com`

Concept identification is based on shallow parsing of the query. A part-of-speech tagger and a chunker (a module that chunks words to phrases) is sufficient for this purpose. To illustrate the independence of the concept language modeling approach to a particular parsing method and compare the effect of syntactic parsing on retrieval, TreeTagger, [66] [2], a decision tree based tagger and chunker for English language was used as alternate parser to parse queries in the retrieval experiments.

4.5 Experiments and Results

Two main questions addressed in the experiments to evaluate the effectiveness of concept language models are: (1) Is the time spent on processing a user query to discover its syntactic structure worth the effort? In other words, How well do concept language model, specifically, concept unigram language model perform in comparison to a smoothed unigram language model? (2) Do the term dependencies discovered through query parsing improve retrieval results over term dependencies learned using co-occurrence statistics? The performance of a concept unigram language model is compared with a smoothed bigram language model for document retrieval.

Experiments were performed on three data sets. The first data set is the Wall Street Journal(WSJ2) sub-corpus of TREC4 from TREC disk 2. The second data set is the TREC4 corpus from TREC disks 2 and 3. Retrieval experiments on both WSJ2 and TREC4 corpus were based on the TREC topics 201-250. The third data set is the TREC8 ad-hoc task corpus with data from TREC disks 4 & 5 and topics 401-450. TREC8 has 528,155

[2] available at `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html`

documents and 2GB of data. For TREC8 data, the title and description of the topics are used as user queries.

The experiments were conducted using the TAPIR toolkit (briefly described in section 3.1.3). TAPIR implements the indexing schemes discussed in Section 4.3. For the experiments, word-based index is created for all the three different data sets. Words are stemmed using a porter stemmer and stop-words are removed before the index is created.

In the experiments, the unigram probabilities are smoothed using Dirichlet-prior with the parameter μ set to a value of 1000. The bigram probabilities are smoothed using unigram probabilities as given in Equation (4.6) with the parameter λ set to 0.1. The unigram probability, $P(t_i|M_D)$ in (4.6) is a Dirichlet-prior smoothed unigram estimate. In Concept Unigram Language Model, the concept probabilities are estimated using their bigram approximations. The bigram estimate weights the empirical bigram document probability at 10% and use Dirichlet-smoothed unigram probability at 90%. The tuning parameters have been selected based on experiments done by other researchers on Dirichlet smoothing [86] and bigram language models [69].

4.5.1 Effect of Concept Identification

This section discusses the experiments comparing the smoothed unigram language model with concept language model. Figure 4.2 plots the precision-recall curves for TREC topics 201-250 on WSJ sub-corpus in TREC disk 2. Table 4.1 shows the corresponding performance measures. The first two columns compare the Dirichlet-prior smoothed unigram language model (SUG) with our approach. The third column gives the percent change in values. Column four gives the significance value according to the one-sided Wilcoxon test. The fifth column gives the number of queries for which the performance improved (I) and the number of queries for which the performance was different (D). The last column gives the significance values according to the one-sided Wilcoxon test on the queries for which the particular statistic for CULM improved over SUG. This column indicates whether the performance improvement is significant if we consider only those queries for which the performance of CULM did improve over SUG. In columns four and six, a star mark after the significance value indicates a statistical significant difference at the 0.05 level.

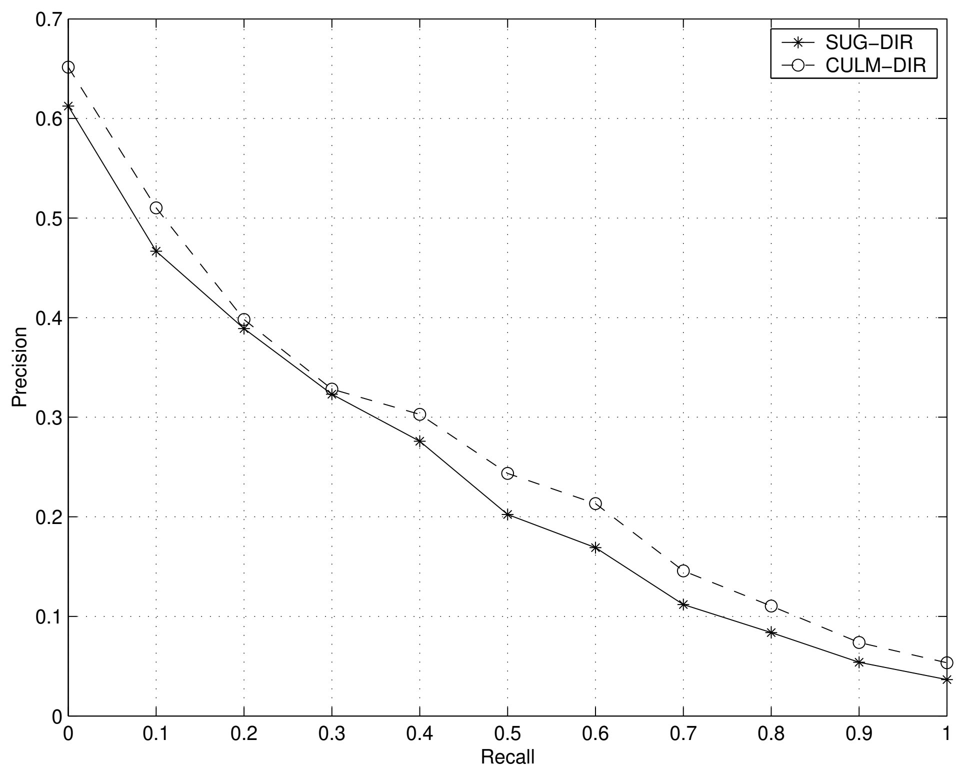

Figure 4.2: Precision-Recall Curve for WSJ2 collection – Dirichlet Prior Smoothing

	SUG-DIR	CULM-DIR	%chg	Wilc.	I/D	I-Wilc.
Rel	1183	1183				
Rret	973	970	-0.3	0.0853	5/12	0.0787
Prec.						
0.00	0.6124	0.6515	6.4	0.0066*	9/17	0.0026*
0.10	0.4667	0.5102	9.3	0.0278*	14/30	0.0023*
0.20	0.3890	0.3981	2.3	0.1116	12/32	0.0297*
0.30	0.3231	0.3282	1.6	0.1882	13/34	0.0349*
0.40	0.2758	0.3028	9.8	0.1767	15/36	0.0159*
0.50	0.2024	0.2435	20.3	0.0110*	17/34	0.0003*
0.60	0.1691	0.2133	26.1	0.0177*	14/33	0.0004*
0.70	0.1119	0.1456	30.1	0.0043*	11/29	0.0002*
0.80	0.0837	0.1104	31.9	0.0005*	11/26	0.0000*
0.90	0.0539	0.0738	36.9	0.0000*	8/20	0.0000*
1.00	0.0365	0.0535	46.6	0.0000*	5/14	0.0000*
Avg	0.2282	0.2541	11.3	0.3008	16/38	0.0212*
Prec.						
5	0.3244	0.3600	11.0	0.0000*	9/13	0.0000*
10	0.2867	0.2978	3.9	0.0000*	9/17	0.0000*
15	0.2607	0.2756	5.7	0.0020*	15/27	0.0000*
20	0.2367	0.2489	5.2	0.0208*	16/31	0.0008*
30	0.1985	0.2089	5.2	0.0036*	13/26	0.0004*
100	0.1162	0.1164	0.2	0.0036*	12/25	0.0003*
200	0.0783	0.0771	-1.5	0.0053*	9/22	0.0017*
500	0.0387	0.0390	0.8	0.0035*	7/16	0.0013*
1000	0.0216	0.0216	0.0	0.0020*	5/12	0.0017*
RPr	0.2395	0.2517	5.1	0.0137*	13/26	0.0021*

Table 4.1: Comparison of Smoothed Unigram Language Model and Concept Unigram Language Model smoothed using Dirichlet priors for WSJ2 corpus.

The concept unigram language model achieves better precision at all levels of recall in the 11-pt recall/precision values and with statistical significance in most of them. While CULM performs at comparable recall values, it provides significant improvement in R-precision value and the precision after N documents. Improvements of 10.1% and 6.2% are observed in the uninterpolated average precision and R-precision values.

From the results given in Table 4.2 for TREC queries 201-250 on TREC4 corpus, similar improvement are observed for all performance measures. The precision-recall plots are given in figure 4.3. Improvements are also observered in the uninterpolated average precision (7.7%), R-precision (4.6%) and recall (2.1%) values. However, not many of the improvements are statistically significant. It is noted that the I/D measure indicates improvement for a number of queries against all performance measures, however it is not spread around to make a number of them statistically significant. In other words there are some queries for which the CULM performs significantly better than SUG. A similar observation was noted by Mitra et al. [47] in their experiments using phrases as index terms. One possible reason for this could be the difference in the corpus size resulting in better probability estimates. For those queries where an improvement in the performance measure was observed, we computed the significance value for the Wilcoxon test and this is given in the last column in Table 4.2. This indicates the improvements that are observed in some queries are indeed statistically significant.

Table 4.3 shows the results for TREC topics 401-450 on TREC8 collection shows performance improvement of CULM over SUG but the improvements are not statistically significant. The I/D column leads us to similar conclusion as observed for TREC4 collection. Figure 4.4 clearly indicates the absence of statistical significance in the improvements observed for TREC8 collection.

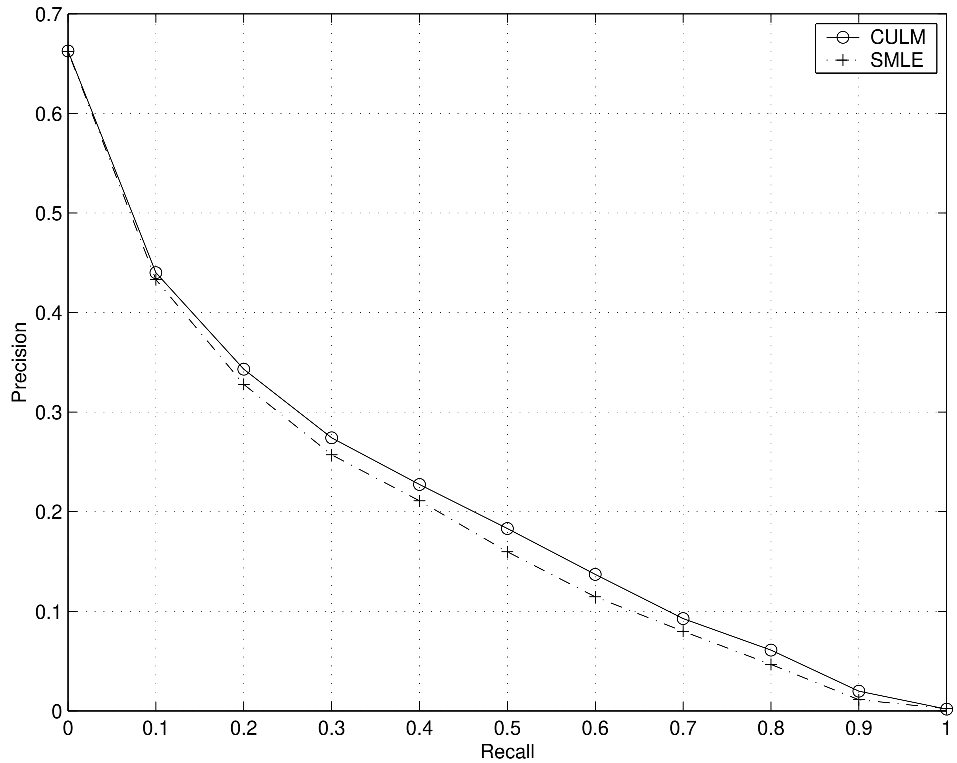

Figure 4.3: Precision-Recall Curve for TREC4 collection – Dirichlet Prior Smoothing

4.5.2 Concept Identification in Shorter Queries

Both title and description fields of the TREC8 topics 401–450 are reported in Table 4.3. The title fields are typically keywords and concepts of interest for the topic, the description provides some more information. The title fields have on the average 2.4 words and represent queries that are typically posed to a web search engine. Table 4.4 compares the performance of the concept language models over smoothed unigram models when only the title fields are used as queries. Different window sizes for concept occurrence for approximating its probability using bigram probabilities were experimented with. The performance in the Table 4.4 corresponds to the window size of 3 words. Concept language models gives a small, but not statistically significant, improvement over smoothed unigram

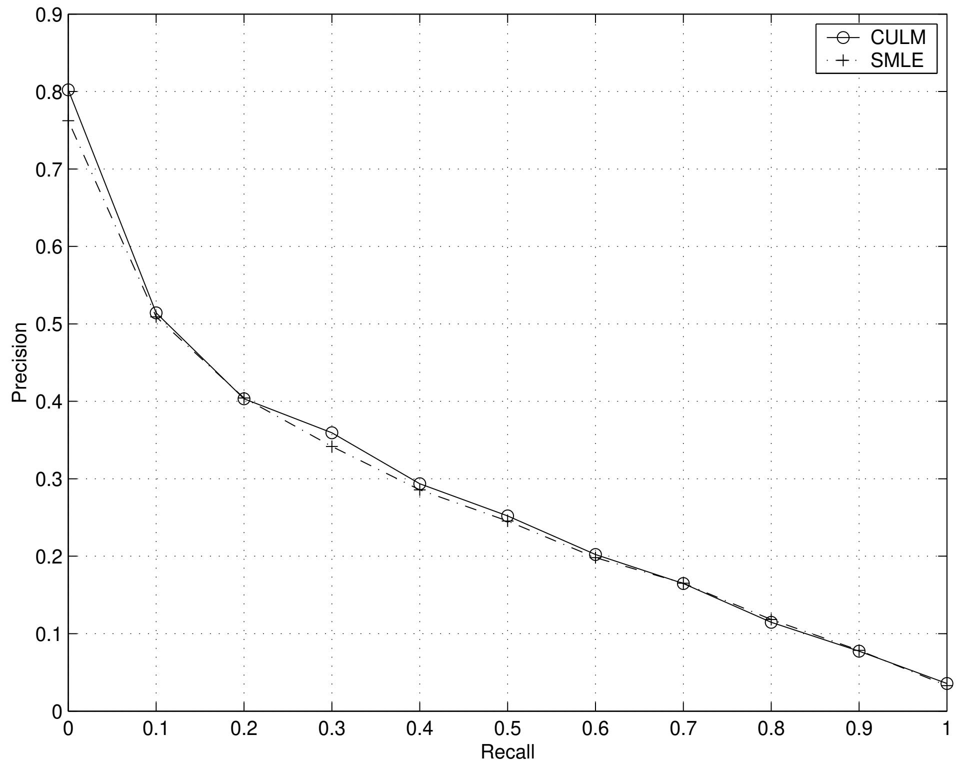

Figure 4.4: Precision-Recall Curve for TREC8 collection – Dirichlet Prior Smoothing

language models. The precision-recall graph is given in Figure 4.5. It is noted that the performance reduces using a bigram language model with title queries.

From our experimental results on three different test collections, we note that using concept unigram language models do improve the retrieval performance over smoothed unigram language models. Improvements are observed on all performance measures, with statistical significance on a number of them.

4.5.3 Concepts vs. Bigrams

In the estimation of concept probabilities in Equation 4.1, the concepts are viewed as sequence of terms and a bigram approximation of the joint term-sequence probability is

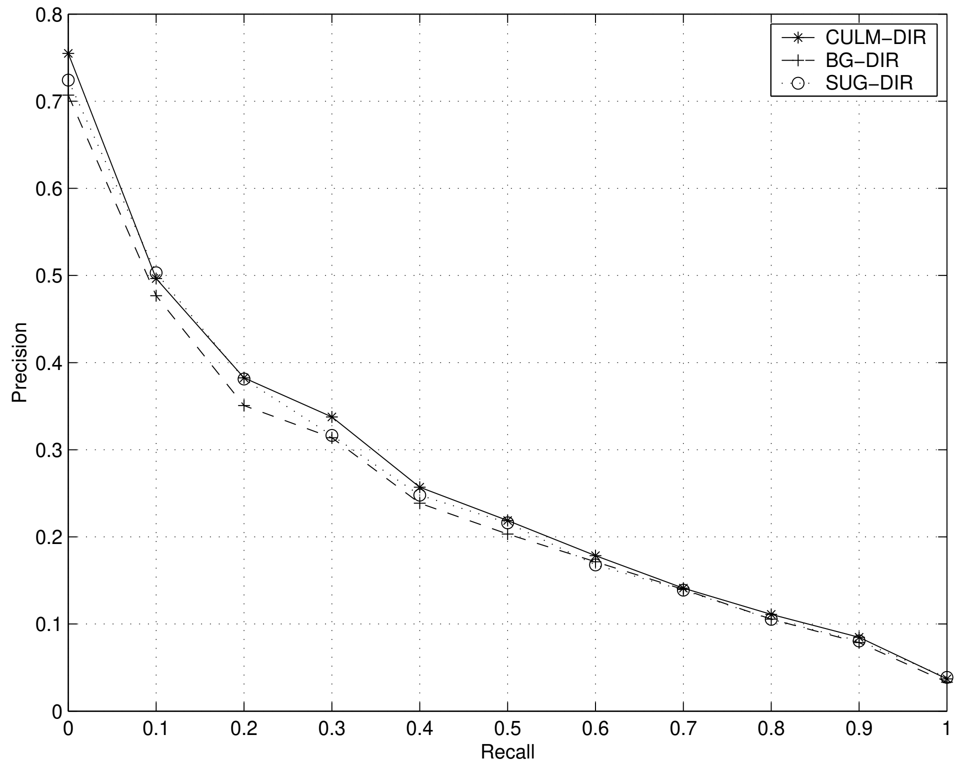

Figure 4.5: Precision-Recall Curve for TREC8 collection using Title Queries – Dirichlet Prior Smoothing

used as an estimate of the concept probability. A multi-word concept can have more than 2 terms and the concept probability, locally, is approximated by a bigram of the terms. Syntactic parsing of the queries identified these concepts. Table 4.5 compares the performance of smoothed bigram language models (SBG) over Concept Unigram Language Model for the three document collections. The statistical significance of these results is evaluate and given in the table for one-sided Wilcoxon test. Concept unigram language models achieve statistically significant improvements over smoothed bigram language models for three of the four test settings. An increase of 4% in average precision is observed in TREC4. It is not statistically significant at the 0.05 level. Statistically significant improvements are observed in the WSJ2 and TREC8 collections. These improvements indicate the effect of concepts identified through syntactic parsing of queries over word-bigrams. Local dependencies captured using bigram for concepts. The dependencies the cross the boundary of a concept or syntactic group do get included in the bigram estimation resulting in lower precision.

4.5.4 Effect of Syntactic Parser

The InfoXtract system was used to parse the queries and identify concepts. InfoXtract is an information extraction engine that includes a shallow parser and can identify named entities and associations in input textual content. However, only the parse tree of the query was used to identify concepts. Is such a sophisticated engine required to identify concepts? Can simple heuristics be used to identify concepts. Since the concepts as assumed to contiguous sequence of words, stop words can be used as delimiters for concepts. For example, the concepts identified using the InfoXtract engine and using stopwords as delimiters are different in the case of TREC topic 231:

- **[InfoXtract]** *Should the [U.S. Government] [provide] [increased support] to the [National Endowment for the Arts]*
- **[StopWords]** *Should the [U.S. Government provide increased support] to the [National Endowment] for the [Arts]*

The identified concepts are placed between square brackets.

Table 4.6 compares the concept language model performance using different parse of the queries with the smoothed unigram and bigram language models. Queries parsed by a syntactic parser perform well the larger collections. The possible difference in training data and tags used in the InfoXtract and TreeTagger can be attributed to the difference in the retrieval performance between the syntactic parsers used. A stopword based concept identification does better on the WSJ2 collection over the retrieval using InfoXtract parsed queries. For the TREC8 collection, using stopwords to identify concepts did not yield good results in comparison to smoothed unigram language model.

Table 4.7 enumerates previously published mean average precision values for the three different document collections and compare them with concept unigram language models. While the details of the models are available in their respective papers, the first column briefly identifies the retrieval model used. The second column indicates whether title and/or description was used as queries. Concept Unigram Language Model performs better than smoothed bigram language models on all three collections. Better performance has been achieved on TREC8 corpus using improved query representation with or without (pseudo-) relevance feedback. These improvements have been achieved - for example in Markov query translation model - by starting from keywords in the query. Better results, we believe, can be achieved by starting from the concepts in the query.

4.5.5 HARD corpus

High Accuracy Retrieval from Documents (HARD) is a new track in TREC that was introduced in 2003. User's information need in *ad hoc* retrieval is represented by query text. In HARD, in addition to the query text, the user has the flexibility to include in the query some metadata to clarify the information need. Figure 4.6 gives an example of HARD topic with associated metadata. The metadata of purpose for this topic is set to

DETAILS and genre set to OVERVIEW indicating that the user is interested in obtaining details on this topic and it should provide a overview of the topic. Other metadata included granularity and familiarity of the user. This section reports the results of using the concept language models for the HARD topics. It is only the query text that is of interested here. Metadata information available about the queries is ignored. The HARD corpus is used here as yet another document collection with truthed queries to demonstrate the efficiency of the concept language models approach.

```
<top>
<num> Number:  001</num>
<title> H1-B Visa Regulations</title>
<descr>
Description:  What kind of legislation has been considered
and adopted regarding the H1B Visa?
</descr>
<narr>
Narrative:  Congressional and media discussion, proposals,
legislation and reaction regarding the H1B Visa are on topic
(including detailed discussions of things like, e.g., rais-
ing the cap on the number of H1B visas for a particular
year).  General discussion of foreign workers in the US,
broader INS regulations and other visa types (e.g., student
visas) are off topic.
</narr>
<hard> item=purpose, value=Details</hard>
<hard> item=genre, value=Overview</hard>
<hard> item=granularity, value=Document</hard>
<hard> item=familiarity, value=3</hard>
</top>
```

Figure 4.6: Sample TREC HARD track topic with query text and metadata

Due to metadata being associated with HARD queries, a document is assigned on of the three values with respect to its relevance to a given query. It is either non-relevant or has soft or hard-relevance for the given query. Soft-relevance corresponds to a document being relevant to the given query and the hard-relevance indicates the document is not only topically relevant but also satisfies the metadata requirements of the query. HARD corpus has 372,219 documents with 1.7GB of data. Soft- and Hard- relevance judgments are available for 50 test topics. The InfoXtract engine was used to parse the queries. The soft-relevance judgments were used in evaluating retrieval performance.

Table 4.8 compares unigram language model smoothed using Dirichlet priors (SUG-DIR) and concept language model (CULM-DIR). Both soft and hard relevance are compared. No evidence other than the query text is used in either retrieval methods. Improvements of 2.7% and 4.2% are observed for hard- and soft-relevance, respectively. Statistical significant improvements for soft-relevance using the concept language models over smoothed unigram language models provides more empirical evidence for the effectiveness of concept language models.

4.6 Summary

A new language modeling approach to information retrieval is proposed in this chapter. The querying process is modeled as a two-layer language model. Language modeling approach using query-likelihood models view a query as a set or sequence of words. In concept language model, a query is viewed as a sequence of concepts and a sequence of words from the query text express the concepts. Some assumptions are made for what defines a concept and how to identify them in a document. In the experiments, a multi-word phrase in the query is viewed as a concept and proximity conditions are used to identify concept occurrence in documents. In concept unigram language model, concept independence assumption is made and the query-likelihood probability for a given document is given by the product of concept-likelihood probabilities. This basic concept language model results improved retrieval performance in different document collections, some of the improvements are statistically significant.

Concept independence is one assumption that can be easily removed and one can generate concept N-grams. However, the interaction between concepts is much more complex to model by simple proximity constraints in documents. While better concept identification in documents will help in improving concept unigram language model performance, it will certainly be useful for exploring higher-order concept models.

The biterm and maximum bigram language models presented in Chapter 3 were based on statistical phrases, specifically, word pairs, identified in queries. In concept language models, syntactic phrases in the query text are identified as concepts of interest to the user. A part of speech tagger and a shallow parser to identify phrases is sufficient for using concept language models. An information extraction system like InfoXtract provides much more than parse tree of the query including named entities and other relations between concepts. While processing documents through an information extraction system will increase index generation time, it takes little time to extract information from user queries. Hence, models that incorporate such extracted information coupled with effective identification of concepts and their relations will improve retrieval performance.

Mitra et al. [47] indicate that phrases by themselves are non-optimal representatives of a query, since normally only a part of the subject matter of a query is covered by phrases. The retrieved set of documents based on phrase matches may include documents that may be about some aspects of the phrase different from that of interest to the user. While the two layer model with concepts independence assumption and bigram approximation for concepts is able to improve retrieval performance, specifically, precision, methods that better capture the semantics of the query need to be explored. For TREC4 query *where are the nuclear power plants in the U.S. and what has been their rate of production* (topic 204), one is really interested in nuclear power plants located in the United States. However, the following phrases, with stop words removed, are the concepts used to represent the query: *nuclear power plants*, *U.S.*, *rate*, *production*. The location restriction is not apparent from the concept sequence. Such information is available in the parse of the query. Methods to capture such semantics in phrase-based retrieval models need to explored to fully exploit the output of NLP on queries.

In the two-layer language models, the concept-likelihood probability was estimated using a bigram approximation of its constituent query terms. These query terms are the set of words or the surface-form the user selected to describe that concept of interest. The user could choose other surface forms or synonyms of the concept and it is likely that a synonym, rather than the user's query text that appears in a document. The concept-likelihood probability, then, can be estimated as the probability of one of the surface-forms or synonyms generated by the document model. The concept language model provides an elegant mechanism for incorporating query expansion in a language modeling approach and in addition, the two-layer approach maintains the mapping of concepts to their synonyms that can be used while estimating the query-likelihood probability. This is one of the future directions for experimentation using concept language models.

Biterm and maximum bigram language models discussed in Chapter 3 and concept language models are based on query-likelihood retrieval models that are restricted by the document representation adopted in the models. A variety of textual and non-textual features contribute to the relevance of a document to a given query. Document-likelihood models that can incorporate such diverse features in estimating the relevance of a document to a given query are presented in the next chapter.

	SUG	CULM	%chg	Wilc.	I/D	I-Wilc.
Rel	6503	6503				
Rret	3276	3345	2.1	0.0889	17/31	0.0105*
Prec.						
0.00	0.6622	0.6627	0.1	0.0454*	11/20	0.0216*
0.10	0.4331	0.4401	1.6	0.4382	13/34	0.1656
0.20	0.3279	0.3432	4.7	0.1499	15/34	0.0244*
0.30	0.2571	0.2742	6.7	0.1271	17/34	0.0089*
0.40	0.2109	0.2273	7.8	0.0358*	13/29	0.0037*
0.50	0.1598	0.1831	14.6	0.0043*	15/25	0.0004*
0.60	0.1146	0.1370	19.5	0.0034*	12/20	0.0007*
0.70	0.0799	0.0926	15.9	0.0016*	9/16	0.0006*
0.80	0.0466	0.0611	31.1	0.0009*	3/7	0.0007*
0.90	0.0111	0.0198	78.4	0.0007*	3/4	0.0006*
1.00	0.0022	0.0018	-18.2	0.0006*	0/1	0.0006*
Avg	0.1876	0.2020	7.7	0.3298	20/41	0.0117*
Prec.						
5	0.4520	0.4560	0.9	0.0113*	11/21	0.0030*
10	0.4100	0.4320	5.4	0.0115*	13/24	0.0041*
15	0.3760	0.4053	7.8	0.0191*	19/30	0.0007*
20	0.3570	0.3790	6.1	0.0206*	17/29	0.0005*
30	0.3253	0.3360	3.3	0.0407*	16/29	0.0068*
100	0.2304	0.2358	2.3	0.1312	15/31	0.0216*
200	0.1710	0.1781	4.2	0.0647	18/35	0.0037*
500	0.1050	0.1064	1.3	0.0868	15/31	0.0102*
1000	0.0655	0.0669	2.1	0.0889	17/31	0.0194*
RPr	0.2475	0.2589	4.6	0.0742	18/34	0.0040*

Table 4.2: Comparison of Smoothed Unigram Language Model and Concept Unigram Language Model where probabilities are smoothed using Dirichlet priors on TREC queries 201–250 on TREC4 corpus.

	SUG	CULM	%chg	Wilc.	I/D	I-Wilc.
Rel	4728	4728				
Rret	2970	2964	-0.2020	0.1823	7/15	0.1224
Prec.						
0.00	0.7623	0.8022	5.2342	0.0779	10/19	0.0308*
0.10	0.5092	0.5143	1.0016	0.5000	16/35	0.1383
0.20	0.4044	0.4032	-0.2967	0.2498	12/34	0.4602
0.30	0.3417	0.3593	5.1507	0.2826	12/29	0.1056
0.40	0.2858	0.2936	2.7292	0.1468	12/26	0.0507
0.50	0.2454	0.2520	2.6895	0.3492	11/28	0.1329
0.60	0.1982	0.2023	2.0686	0.1383	11/24	0.0534
0.70	0.1651	0.1647	-0.2423	0.1150	8/19	0.0700
0.80	0.1182	0.1147	-2.9611	0.1556	7/16	0.1224
0.90	0.0784	0.0774	-1.2755	0.1873	4/10	0.1586
1.00	0.0328	0.0356	8.5366	0.1616	2/3	0.1586
Avg	0.2610	0.2638	1.0728	0.3631	22/42	0.0494*
Prec.						
5	0.5000	0.5080	1.6000	0.1993	14/24	0.1011
10	0.4500	0.4480	-0.4444	0.1823	13/24	0.1126
15	0.4253	0.4293	0.9405	0.2954	11/20	0.1924
20	0.3990	0.4080	2.2556	0.1758	13/21	0.1056
30	0.3560	0.3600	1.1236	0.2680	10/22	0.1526
100	0.2232	0.2320	3.9427	0.1468	15/27	0.0369*
200	0.1613	0.1664	3.1618	0.0752	12/24	0.0143*
500	0.0971	0.0988	1.7508	0.0501	9/21	0.0217*
1000	0.0594	0.0593	-0.1684	0.0369*	7/15	0.0196*
RPr	0.3040	0.3114	2.4342	0.2247	13/25	0.0692

Table 4.3: Comparison of Smoothed Unigram Language Model and Concept Unigram Language Model where probabilities are smoothed using Dirichlet priors for TREC8 corpus.

Measure	SUG-DIR	CULM-DIR	% change	Wilc.	I/D	I-Wilc.
AvgP	0.2421	0.2464	1.8	0.2080	20/31	0.0256*
Recall	2795/4728	2768/4728	-1.0	0.0821	5/10	0.0669
RPrec	0.2844	0.2933	3.1	0.2750	14/21	0.0573
InitPr	0.7242	0.7548	4.2	0.2621	9/16	0.2025

Table 4.4: Comparison of Smoothed Unigram Language Model and Concept Unigram Language Model for TREC8 Title queries

Collection	Measure	SBG	CULM	% change	Wilc.	I/D	I-Wilc.
WSJ2	Avg	0.2442	0.2541	4.1	0.0293*	28/39	0.0002*
	Recall	961/1183	970/1183	0.9	0.0003*	4/7	0.0003*
	RPrec	0.2475	0.2517	1.7	0.0135*	11/20	0.0043*
	InitPr	0.6005	0.6515	8.5	0.0018*	16/20	0.0002*
TREC4	Avg	0.1938	0.2020	4.2	0.0658	30/46	0.0014*
	Recall	3289/6503	3345/6503	1.7	0.3987	15/32	0.1925
	RPrec	0.2516	0.2589	2.9	0.0835	24/35	0.0072*
	InitPr	0.6631	0.6627	-0.1	0.0958	15/25	0.0344*
TREC8	AvgP	0.2538	0.2638	3.9	0.0014*	33/46	0.0000*
	Recall	2900/4728	2964/4728	2.2	0.0000*	12/14	0.0000*
	RPrec	0.3028	0.3114	2.8	0.0023*	16/25	0.0002*
	InitPr	0.7393	0.8022	8.5	0.0006*	12/18	0.0002*
TREC8 Title only	Avg	0.2294	0.2453	6.9	0.0017*	17/22	0.0002*
	Recall	2354/4728	2771/4728	17.7	0.0003*	6/7	0.0002*
	RPrec	0.2799	0.2911	4.0	0.0043*	10/15	0.0008*
	InitPr	0.7070	0.7528	6.5	0.0013*	12/15	0.0002*

Table 4.5: Comparison of Smoothed Bigram Language Model and Concept Unigram Language Model

Collection	Measure	SUG-DIR	BG-DIR	InfoXtract	Stopwords	TreeTagger
WSJ2	AvgP	0.2282	0.2442	0.2541	0.2582	0.2603
	Recall	973/1183	961/1183	970/1183	956/1183	951/1183
	RPrec	0.2395	0.2475	0.2517	0.2666	0.2652
	InitPr	0.6124	0.6005	0.6515	0.6289	0.6217
TREC4	AvgP	0.1876	0.1938	0.2020	0.1980	0.2042
	Recall	3276/6503	3289/6503	3345/6503	3239/6503	3206/6503
	RPrec	0.2475	0.2516	0.2589	0.2578	0.2568
	InitPr	0.6622	0.6631	0.6627	0.6676	0.6692
TREC8	AvgP	0.2610	0.2538	0.2638	0.2560	0.2601
	Recall	2970/4728	2900/4728	2964/4728	2929/4728	2729/4728
	RPrec	0.3040	0.3028	0.3114	0.3100	0.3154
	InitPr	0.7623	0.7393	0.8022	0.7569	0.7797

Table 4.6: Performance comparison for concept language models using different parsing of the queries and the smoothed unigram and bigram language models.

Retrieval Model/Method	Query	AvgP
Collection: WSJ2		
Concept Unigram Language Model	Desc.	0.2512
Song and Croft [69]: Smoothed Bigram Language Models	Desc.	0.2359
Srikanth and Srihari [75]: Biterm Language Models	Desc.	0.2393
Collection: TREC4		
Concept Unigram Language Model	Desc.	0.2020
Song and Croft [69]: Smoothed Bigram Language Models	Desc.	0.1917
Collection: TREC8		
Concept Unigram Language Model	Title+Desc.	0.2638
Lafferty and Zhai [37]: Markov Chain query translation model	Title	0.266
Lafferty and Zhai [37]: Markov Chain query translation model conditioned with Pseudo-feedback	Title	0.294
Narita and Ogawa [49]: Unigram and phrasal terms with proximity based weighting	Title	0.2573
Narita and Ogawa [49]: Unigram and phrasal terms with proximity based weighting	Title+Desc.	0.2725
Zargoza et al. [83]: Combining Bayes-Predictive and Linear Interpolation Smoothing	Title+Desc.	0.257

Table 4.7: Comparison of Concept Unigram Language Models to previous best performing systems on different document collections

Collection	Measure	SUG-DIR	CULM-DIR	% change	Wilc.	I/D	I-Wilc.
Hard-Rel	AvgP	0.2904	0.2981	2.7	0.0510	18/30	0.0001*
	Recall	3591/5123	3548/5123	-1.2	0.0101*	13/18	0.0040*
	RPrec	0.3230	0.3286	1.7	0.0402*	12/21	0.0101*
	InitPr	0.6806	0.6822	0.2	0.1164	5/13	0.0841
Soft-Rel	AvgP	0.3396	0.3540	4.2	0.0156*	19/31	0.0001*
	Recall	5100/7576	5069/7576	-0.6	0.0064*	13/19	0.0021*
	RPrec	0.3719	0.3784	1.7	0.0124*	14/23	0.0008*
	InitPr	0.7840	0.7922	1.0	0.0056*	6/12	0.0028*

Table 4.8: Comparison of Smoothed Unigram Language Model and Concept Unigram Language Model for HARD corpus

Chapter 5

Maximum Entropy Language Models for Information Retrieval

This chapter introduces a new approach to document retrieval by estimating directly the document-likelihood probability using maximum entropy models. A number of features, beyond term occurrence and co-occurrence, contribute to the relevance of a document to a given query. Section 5.1 motivates the need for models that incorporate such diverse information towards estimating relevance. A similar problem in speech recognition is solved using the maximum entropy principle. Section 5.2 briefly reviews the application of statistical language models in speech recognition, the maximum entropy models that incorporate diverse information sources for estimating the word-prediction probability and the whole sentence maximum entropy model that directly estimates the sentence likelihood probability.

In the approach proposed in this chapter relevance status value of a documents is given by the document likelihood probability of relevance. Document likelihood models that are related to the traditional probabilistic retrieval models is presented in section 5.3. The whole document maximum entropy model is presented in section 5.4 followed by related work on using diverse set of features for retrieval and the use of maximum entropy principle for IR. In maximum entropy models, constraints defined on the probability distribution are as important as the event space over which the distribution is defined. In whole document maximum entropy model the constraints are information sources for relevance to given query. The features or information sources used in the experiments presented in this chapter are discussed in section 5.6. Experimental results on standard TREC document collection is presented with some observations in section 5.7.

5.1 Motivation

In language modeling approach to information retrieval using smoothed unigram language models, the relevance status value of a document for a given query is estimated based on the probability distribution defined on the vocabulary space. The event space of term co-occurrence has been explored in N-gram language models and the language models proposed in Chapters 3 and 4. However, a number of statistical and semantic features beyond term occurrence and co-occurrence statistics contribute to the relevance of a document to a given query. Statistical clues include number of matched query terms, overlap with prior queries from the user, overlap with prior queries addressed to the document collection, and document length. Semantic features include possible topicality of the query and documents, sense of query words. Including such features in a model requires changes to the event space and every additional feature will require a new retrieval or ranking function to be defined in language modeling approach to IR.

The graphical model corresponding to query-likelihood probability as estimated in a unigram language model is given in figure 5.1(a). The probability of query term q_i depends on the document model, M_d. The bigram language model is given in figure 5.1(b) where the probability of query term q_i depends on the document model, M_d and the previous query term q_{i-1}. The use of additional information in the query-likelihood models requires changes to the estimation model as in the case of moving from a unigram to a bigram language model.

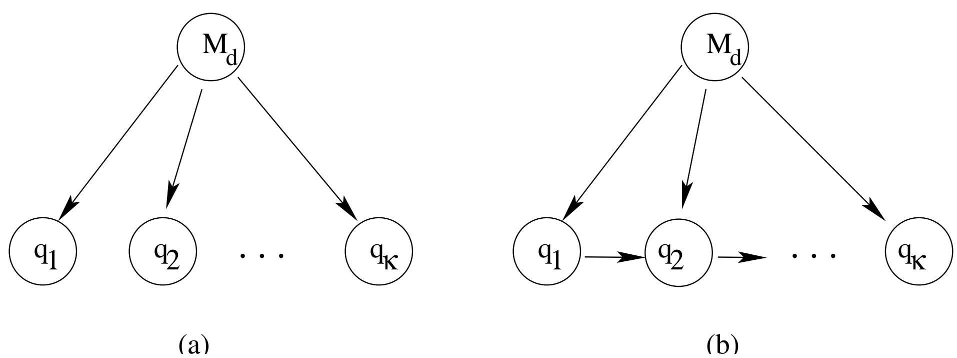

Figure 5.1: Graphical Model for query likelihood estimation (a) unigram Model, (b) bigram Model

The query-likelihood probability estimation is based on the chain rule of probabilities and its approximations. Some of the statistical and semantic clues given above can, at best, be incorporated in the chain rule indirectly to effect their contribution to query-likelihood probability estimation. A simple method adopted in some applications like search personalization is to use such features as filters for the retrieval results. For example, a document that matches the topics of interest as in an user profile can be assigned a higher relevance value than one that satisfies the query, but not the user's preference on topics. Instead of adhoc weighting methods to incorporate such preferences, one is interested in models that can incorporate such diverse information on relevance to a given query in estimating the relevance status value of a document and do not force changes to the ranking or probability estimation process.

A similar problem in speech recognition has been solved using maximum entropy models. Section 2.3 presented the application of statistical language models for speech recognition including the different solutions based on maximum entropy principle. For completeness, the next section briefly outlines the use of a variety of syntactic and semantic features for the sentence-likelihood estimation in speech recognition, including the maximum entropy model for combining these diverse features and the whole sentence maximum entropy model that directly estimates the sentence-likelihood probability. A similar approach to document retrieval in section 5.4

5.2 Whole Sentence Maximum Entropy Models

This section briefly reviews the application of statistical language modeling to speech recognition. In speech recognition, statistical language models are used to select one among the many candidate sentences predicted from the acoustic model for a spoken utterance. The probability of an utterance or sentence, $s = w_1 w_2 \ldots w_n$ is estimated by decomposing it using the chain rule as product of conditional distributions

$$P(s) = \prod_{i=1}^{n} P(w_i|w_1, \ldots, w_{i-1}) = \prod_{i=1}^{n} P(w_i|h_i) \tag{5.1}$$

where h_i is the history for predicting w_i. Different techniques have been explored to estimate $P(w_i|h_i)$. An N-gram model, specifically, a smoothed trigram model, seems to be a good approximation for speech recognition. Smoothed trigram language model is estimated using a combination of trigram, bigram and unigram statistics from some training data.

A variety of features, like N-grams, word triggers, have been shown to be useful for generating better language models. Each of these features provide some information towards the recognition task. Identifying that the use of linear interpolation to combine these features make sub-optimal use of these features, Rosenfeld [62] proposed maximum entropy models as alternate method for combining such diverse information sources for predicting the next word. Each information source is modeled as a constraint the word prediction probability distribution should satisfy and the most uniform among them is selected as the maximum entropy distribution. While subsequent publications have incorporated syntactic and semantic clues [36] and topic of discourse [80] in sentence prediction, underlying models are modified to incorporate these additional features. Non-textual clues like sentence length that have been shown to be useful in speech recognition cannot be incorporated directly in the above model [64].

Rosenfeld [63] proposed whole sentence maximum entropy model which is a maximum entropy model, that directly estimates the probability of whole sentence or utterance. The probability of a sentence, s is given by

$$P(s) = \frac{1}{Z}P_0(s)exp(\sum_i \lambda_i f_i(s)) \tag{5.2}$$

where the $\{\lambda_i\}$'s are the parameters of the model, Z is a normalizing constant that depends only on the parameters and $\{f_i(s)\}$ are arbitrary computable features of the sentence s. $P_0(s)$ is prior or initial distribution that can be uniform or obtained by alternate formalism (example, by using chain rule and conditional distributions like N-grams). An exponential model (any expression of the form (5.2)) is a maximum entropy model when its parameters are chosen such that the distribution satisfies certain constraints. The constraints typically specify an expectation value,K_i for each feature $f_i(s)$:

$$E_p[f_i] = K_i. \tag{5.3}$$

The value K_i is computed as the expectation value of the feature under the empirical distribution, $\tilde{P}$, of some training sentences $\{s_1, \cdots, s_N\}$. Among distributions which satisfy the constraints, the exponential distribution that are closer (in the Kullback-Leibler divergence sense) to the prior distribution is assigned to $p(s)$. If the prior distribution is uniform, the resultant distribution is a maximum entropy distribution.

Iterative Scaling methods have been explored for solving the entropy optimization problem. Specifically, Generalized Iterative Scaling [20] is a simple procedure that starts with arbitrary parameter (λ_i's) values and adjusts the parameters values by comparing the expected values for each feature against its target value estimated using $\tilde{P}$. The update rule for GIS at kth iteration for parameter λ_i is given by

$$\lambda_i^{k+1} = \lambda_i^k + \frac{1}{F_i} log \frac{E_{\tilde{P}}[f_i]}{E_P[f_i]} \tag{5.4}$$

where F_i is the step size. Improved Iterative Scaling (IIS) [22], limited memory variable metric methods [53] methods have been explored to solve the entropy optimization problem. In our experiments, we used our implementation of the GIS method.

The direct estimation of sentence probabilities using (5.2) involves two computational infeasible tasks that require approximation methods. The expectation $E_P[f_i] = \sum_s P(s)f_i(s)$ involves summation over *all* possible sentences s, an infeasible enumeration task. Sampling techniques have been proposed to approximately compute the expectation values. The second infeasible task is of estimating the normalizer Z which is a summation over all possible sentences, $s : Z = \sum_s P_0(s)exp(\sum_i \lambda_i f_i(s))$. This constant however, is not necessary while learning the parameters by sampling methods. It can also be ignored in applications like speech recognition where one in interested in relative ranking of the candidate utterances.

Chen and Rosenfeld [14] compared different sampling techniques for estimating the expectation values. They compared the Metropolis sampling [45], independence sampling (a variant of Metropolis sampling) and importance sampling techniques. Metropolis sampling starts with a randomly chosen initial sentence. For each word position in the sentence, a new word is proposed to replace it and the change is accepted with some probability. After all word positions have been visited, the final modified sentence is added to the sample. Metropolis sampling can be modified to select units larger than a single word to be replaced in each iteration. In independence sampling, the complete sentence can be replaced based on a probability distribution $Q(s)$ that is similar to the sampled distribution $P(s)$. In importance sampling, a sample $S' = \{s_1, s_2, \ldots, s_M\}$ generated according to the distribution $Q(s)$ that is similar to $P(s)$ is used to estimate the expectation value. Each sample s_j is counted $P(s_j)/Q(s_j)$ times resulting in the expected value

$$E_P(f_i) \approx \frac{\sum_{j=1}^{M} \frac{P(s_j)}{Q(s_j)} f_i(s_j)}{\sum_{j=1}^{M} \frac{P(s_j)}{Q(s_j)}} \tag{5.5}$$

The independence and important sampling methods have been shown to perform better than Metropolis sampling for the whole sentence maximum entropy models [14]. A sampling method similar to importance sampling is used in the estimating whole document maximum entropy models for document retrieval described in section 5.4.

While approximation techniques are employed in the estimation of sentence probability, the whole sentence maximum entropy language model facilitates global sentence information and influences from preceding sentences/dialogue to be efficiently encoded in the model [64]. This method for modeling and estimating sentence probabilities is shown in Section 5.4 to yield well to the setting of document likelihood ranking for retrieval. The document-likelihood model for retrieval and a particular language modeling approach, relevance-based language models, are discussed next.

5.3 Document Likelihood Models

In probabilistic retrieval models, ranking is based on the probability of relevance, $P(R|Q,D)$ given the query and document. Using Bayes rule, this probability can be expressed as

$$P(R|Q,D) = \frac{P(Q,D|R)P(R)}{P(Q,D)} \tag{5.6}$$

where $P(R)$ is the prior probability of relevance, $P(Q,D)$ is the probability of a random document begin relevant to the given query and $P(Q,D|R)$ is the probability of the conditional probability of document, D, being relevant to query, Q. In the above, the probability relevance given the query and document can be assumed to be proportional to $P(Q,D|R)$. Lafferty and Zhai [38] show that the classical probabilistic retrieval models and the language modeling approach can be derived from $P(Q, D|R)$ by expanding it in two different ways. Language modeling approach as proposed by Ponte and Croft [56] corresponds to expanding $P(Q, D|R) = P(Q|D, R)P(D|R)$. The alternate expansion leads to document-likelihood probability: $P(Q, D|R) = P(D|Q, R)P(Q|R)$. This is used in the probabilistic retrieval models (ref. section 2.1.2) and the relevance-based language models. To discount common aspects of relevant and non-relevant documents, the document likelihood ratio ($P(D|Q, R)/P(D|Q, \tilde{R})$) is used in these models. However, the proposed whole document maximum entropy model directly estimates the document likelihood probability $P(D|R)$.

Lavrenko and Croft [39] proposed relevance-based language modeling approach that, given a query, ranked documents based on the document likelihood ratio. Given a model of relevance, R, and viewing document as a sequence of terms, the document likelihood ratio is estimated as

$$\frac{P(D|R)}{P(D|\tilde{R})} = \prod_{w \in D} [\frac{P(w|R)}{P(w|G)}]^{n(w,D)} \tag{5.7}$$

where $P(w|R)$ is the probability of selecting the word, w, from the relevance model. The corpus model, G, approximates the non-relevance model,$\tilde{R}$ and $n(w, D)$ is the occurrence count of the word w in document D. The corpus model can be generated from corpus statistics.

In document retrieval, only the document collection and the query are available. No prior knowledge of relevance is available to estimate the relevance-model, $P(w|R)$. Lavrenko and Croft [39] propose a method for estimating the relevance model using a set of highly-relevant documents for the given query. To estimate the relevance model, they observe that the query Q should also be generated by the relevance model and hence the probability $P(w|R)$ can be approximated to $P(w|Q)$ which corresponds to the probability of selecting the word w after the selection of the query terms Q. Using Bayes rule, this approximates to

$$P(w|R) \approx P(w|Q) = \frac{P(Q|w)P(w)}{\sum_v P(Q|v)P(v)} \tag{5.8}$$

Making the assumption that the terms in the query Q are generated independently given the word w, the probability $P(Q|w)$ is given by

$$P(Q|w) = \prod_q P(q|w) \tag{5.9}$$

Using the models, M, of highly-ranked relevant documents, the relevance model is derived as

$$P(w|R) \approx P(w) \prod_q (\sum_M P(q|M)P(w|M)\frac{P(M)}{P(w)}) \tag{5.10}$$

where $P(q|M)$ and $P(w|M)$ are the word probability from the document model M. The probability $P(w)$ of randomly selecting word w is estimated from the highly-relevant documents for the given query.

Relevance-based language models perform some form of "semantic smoothing" resulting in improved retrieval performance. The term co-occurrence information is used in estimating the relevance model. Any new information to generate a better relevance models would require a significant change in the estimation procedure. For example, in concept language models (chapter 4), a query, Q, was viewed as a sequence of concepts, $c_1, c_2, \ldots, c_k$. Given such a representation, the query probability given word w in 5.9 can be written as

$$P(Q|w) = \prod_i P(c_i|w) \tag{5.11}$$

Such an approximation could result in better relevance model than the query term independence assumption in (5.9). Incorporating concepts in estimating relevance model, however, requires changes to the model generation and estimation procedures. Incorporating other syntactic and semantic features would require precise modeling and estimation functions.

The relevance model in equation (5.10) uses a set of highly-ranked documents and co-occurrence of terms with query words to estimate the probability distribution. These can be viewed as a features exploited by the relevance model. Each feature contributes to estimating relevance of a document for a given query; The estimation task, then, becomes one of combining these information sources of relevance. The whole document maximum entropy model proposed in the next section estimates the document likelihood probability by modeling the information sources of relevance as constraints that the document likelihood probability should satisfy.

5.4 Whole Document Maximum Entropy Models for IR

This section presents the whole document maximum entropy model that takes its name from a similar approach for speech recognition discussed in Section 5.2. Given a query, Q, a relevance or query model, R, is associated with it. This, in some sense, represents the properties of an "ideal" relevant document to the query. Unlike the relevance-based language model proposed by Lavrenko and Croft [39] that is defined on the set of terms or the vocabulary space, the relevance model, R, here refers to a set of properties that are computable functions or features,$\{f_1(d), f_2(d), \ldots, f_k(d)\}$, of a document that characterize or contribute to the relevance of a document to given query. A document is viewed as a "bag of features". Documents are ranked based on their document likelihood probability, $P(d|R)$ given by

$$P(d|R) = \frac{1}{Z} P_0(d|R) exp(\sum_i \lambda_i f_i(d)) \tag{5.12}$$

The features $\{f_i(d)|i = 1, 2, \ldots, m\}$ are arbitrary computable features of a document. The parameter λ_i is the weight of the feature f_i. Z is a constant that normalizes the probabilities and depends only on the parameters λ_is.

The constraints to estimate the maximum entropy distribution are based on the expectation values of the features. Training data in the form of a set of highly ranked documents for the given query are used in defining the constraints.

$$E_p[f_i] = \sum_d p(d|R) f_i(d) = K_i \tag{5.13}$$

While the conditional probability, $P(d|R)$ is used in the Equation (5.13), the distribution that is being learned is that of an "ideal" relevant document for the given query. The elegance of the model is that features that contribute to relevance are introduced as constraints that the document likelihood probability should satisfy. Such features are not restricted to be selected based on the current query and its relevance. Information from prior queries can be used in these constraints. The experiments reported here, however, use only the features from a set of highly-relevant documents for the current query.

As with the whole sentence maximum entropy models for speech recognition, the whole document maximum entropy models also have to deal with the infeasible task of computing the expectation over *all* documents in the iterative scaling solution and estimating the normalizer, Z, in equation (5.12). The normalizer, Z, being a constant, can be ignored in document retrieval application, as one is interested in the relative ranking of documents being relevant to the given query. As mentioned in section 5.2, the importance sampling method is used in the experiments. The importance sampling method is based on availability of samples (documents, here) that exhibit similar distribution to that of the estimated document-likelihood probability.

For the purpose of modeling document-likelihood probability as a whole document maximum entropy model, a set of highly-relevant documents is required for defining the constraints and a sample of documents with distribution similar to $P(d|R)$ is required approximately computing the expectation values of the estimated probability distribution. One option for identify the highly-relevant document set is to use the top N (typically, 50 or 100) documents as retrieved by a baseline retrieval system. Lavrenko and Croft [39] use the top 100 documents retrieved using smoothed bigram language model to estimate the relevance model for a given query. As a sample of documents that has "similar" likelihood distribution as the distribution that is being estimated, the top $M(>> N)$, can be used as a sample for computing the expectation values. In the experiments, the document sample is set to the top 500 documents retrieved using one of the best performing retrieval models – smoothed unigram language models using Dirichlet priors. While this option ties the performance of the whole document maximum entropy model to the baseline retrieval system, other sampling techniques require more computation cycles in solving the maximum entropy problem.

5.5 Related Work

Some prior work in language modeling approaches and other retrieval models is discussed in Chapter 2. This section compares the proposed approach with some of those models.

The whole document maximum entropy model ranks documents based on the document-likelihood probability, $P(D|R)$. Binary independence models [58] and relevance-based language models [39] approach use the odds of relevance or document-likelihood ratio ($P(D|R)/P(D|\tilde{R})$) for ranking documents. This ratio suppresses some common aspects like document length in relevance estimation and can be computed using the relevance model as in Equation (5.10). The use of chain rule in estimating document likelihood in the relevance-based model approach restricts the kinds of information sources that can be included in the estimation process. As indicated in section 5.3, smoothed probability estimates are used for computing the probabilities $P(w|M_d)$ and $P(q|M_d)$. However, incorporating other clues to generate a better relevance model requires changes to the underlying estimation method.

In description-oriented probabilistic models, the odds of relevance is estimated directly using the linear [25] or logistic regression [17, 13]. The number of query term matches is used in some probabilistic retrieval methods [17, 13] and has been shown to be useful for estimating relevance. Intuitively, a document with higher query term matches can be expected to be more relevant than one with lower query term matches. Some semantic information, like topicality of the document or the possible topics of a given query, cannot be incorporated in models using the event space based on vocabulary of terms.

The whole document maximum entropy model can be viewed as a logistic model for the document likelihood probability. Cai et al [9] make similar observation for whole sentence language models for speech recognition (ref. section 2.3.2). The exponential form of document likelihood probability in equation (5.12) can be written as a logistic function with appropriate parameters:

$$log(P(d)) = \alpha_0 + \sum_i \alpha_i f_i(d) \tag{5.14}$$

Define a random variable Y such that

$$Y = \begin{cases} 1 & \text{if } d \in P \\ 0 & \text{if } d \in P_0 \end{cases}$$

where $d \in P$ means the relevance of the document to given query is captured by the probability distribution $P(d|R)$ and $d \in P_0$ means the baseline retrieval system's relevance assignment $P_0(d|R)$ best represents the relevance of the document to the given query. A value of 0 for the random variable Y indicates that the additional features ,f_i, are not significant to be considered for relevance of document to given query. A new function $h(d) = P(Y = 1|d)$ is defined and by Bayes theorem

$$h(d) = P(Y = 1|d) = \frac{P(d|Y = 1)P(Y = 1)}{P(d|Y = 1)P(Y = 1) + P(d|Y = 0)P(Y = 0)} \tag{5.15}$$

Assuming uniform distribution for prior probability of Y,

$$\frac{h(d)}{1 - h(d)} = \frac{P(d)}{P_0(d)} \tag{5.16}$$

using the exponential form of $P(d)$ in Equation (5.12) and taking logarithms gives

$$\begin{aligned} \log(\frac{h(d)}{1 - h(d)}) &= -\log Z + \sum_i \lambda_i f_i(d) \\ &= \beta_0 + \sum_i \beta_i f_i(d). \end{aligned} \tag{5.17}$$

The Equation (5.17) has the form of logistic regression, when $f_i(d) = x_i$ and the sentence probability can be written as

$$\hat{P}(d) = P_0(d)exp(\hat{\beta}_0 + \sum_i \hat{\beta}_i x_i) \tag{5.18}$$

The odds of the random variable Y estimated using this logistic model, corresponds to the odds that the document likelihood probability using additional clues captures the relevance of the document over the baseline estimate.

A parallel can be drawn between the approaches discussed in this dissertation and the association between model-oriented and description-oriented approaches in probabilistic retrieval models (ref. Section 2.1.2. The model-oriented approaches (for example, BIR models) make some probabilistic assumptions on the terms used for representing documents and queries and rank documents based on the document-likelihood ratio. The ranking function used depends on the selection of document and query representations. A change in the representation, accordingly, changes the ranking model used. In description-oriented approaches (for example, Fuhr model, staged logistic regression) a set of descriptors or features are associated with relevant query-document pairs and these features are used to estimate the odds of relevance of a document. In the logistic regression model, prior queries and their relevance judgments are used to learn the feature weights. The whole document maximum entropy model can be viewed as a description-oriented, language modeling approach for information retrieval. A significant difference between the approach proposed here and the description-oriented probabilistic approaches is that documents are ranked based on document-likelihood probability, $P(d|R)$ rather than the odds of document relevance $P(d|R)/P(d|\tilde{R})$. The whole document maximum entropy models are learned from relevant or highly-relevant documents only.

The use of maximum entropy principle for information retrieval has been suggested by Cooper [16] to address some of the unrealistic simplifying assumptions concerning statistical independence in probabilistic retrieval models. Identifying the shortfalls of boolean and weighted request ranking, Cooper illustrated that maximum entropy principle can be applied to estimate the usefulness of a document to given weighted query where the user-assigned weight for a term is viewed as the fraction of documents indexed by the term that the user will find useful. No empirical results, however, were presented. Kantor and Lee [33] proposed a method for using the maximum entropy model for retrieval where a description vector is defined for each document and constraints are based on prior information on the expected distribution of relevant documents. The probability that a document characterized by an individual description will be relevant (or non-relevant) is assumed to be available to defined the constraints.

Lee and Kantor [40] explored some variants of the maximum entropy models that handle possible inconsistences between estimated probabilities and actual data. They showed in some toy numerical examples that maximum entropy principle makes sensible decisions on relevance of documents with frequently co-occurring terms. As a large scale experiment, Kantor and Lee [34] explored maximum entropy principle in the TREC5 routing task. In Routing task, a stream of documents is filtered for fixed information need. Relevance information is available from the user for documents deemed relevant by the system. The maximum entropy solution used both relevant and non-relevant information provided in the feedback.

The whole document maximum entropy model proposed in this chapter uses the maximum entropy principle and applies it for the ad-hoc retrieval task. The following points highlight the features of the model and identify the differences from prior approaches:

- The information used to generate the maximum entropy model is based on highly-relevant documents only. Corpus information is used to regularize the estimated model as maximum entropy models are susceptible to over fitting. No assumption is made on the relevance of other documents in the corpus. In other words, non-relevance information in the form of documents that are deemed not relevant to the given query is not used the estimating likelihood probabilities.

- Earlier discussions on the application of maximum entropy principle to retrieval have represented documents as a "description-vector" and the descriptions are term-based. Limited empirical results available on large document collections. Textual and non-textual features can used to describe relevant documents in the proposed approach.

- Description-oriented probabilistic models – those using linear or logistic regression – have explored functions defined on term statistics as features that describe the odds of relevance of a document to given query. The whole document maximum entropy model directly estimates the document likelihood probability using relevance information. Syntactic and semantic features can be incorporated in the model by identifying them as additional constraints that the document likelihood probability should satisfy.

Greiff and Ponte [29] show that traditional probabilistic IR models can be reproduced using the maximum entropy principle. Using particular set of constraints, they derive the binary independence model [58] and the combination match model [19] using maximum entropy principle. The event space corresponds to the observation of a document with respect to a given query, $\Omega = \mathbf{X} \times \mathbf{R}$. Associated with each observation is a set of binary random variables representing s query terms, $X_1, X_2, \ldots, X_s$ and relevance, R. The constraints are defined on the $P(X_i|R)$ and $P(R)$. In the proposed approach, the document likelihood probability, $P(\mathbf{X}|\mathbf{R})$ is directly estimated as maximum entropy distribution. The constraints are given by feature functions corresponding to relevance to given query.

Language modeling approaches reduced the problem of information retrieval to that of probability estimation and smoothing. The implicit assumption is that the representation of the language models is based on terms or words in the vocabulary and their interactions. Relevance of a document to a given query can be based on features beyond term occurrence and co-occurrence like matched term counts, topicality of the document and user preferences. some of the features used in the retrieval experiments are presented next.

5.6 Information Sources for Relevance

A number of features that contribute towards the relevance of a document to a given query are modeled as constraints in the whole document maximum entropy models.

The set of features that are explored in the experiments are presented in this section. These features, by no means a comprehensive list, are selected based on the availability of such information in the test collections used in the experiments. The intent here is to demonstrate that the whole document maximum entropy model is an effective framework for incorporating diverse and not necessarily independent features for document retrieval. The results and observations of the experiments on standard document collections are presented in section 5.7.

5.6.1 Query Term Hits Feature

One set of features is based on the query terms. Given a query, $Q = q_1, q_2, \ldots, q_k$, a constraint can be defined based on each one of the query terms with the feature value

$$f_i(d) = f_q(d) = n(q, d) \tag{5.19}$$

where $n(q, d)$ is the number of occurrences of the query term q in document, d. The corresponding constraint is given by

$$\sum_{d} p(d|R) f_q(d) = \sum_{d \in R_Q} \tilde{p}(d|R) f_q(d) = K_q \tag{5.20}$$

where R_Q is the set of highly-relevant documents for the given query. Maximum entropy models overfit the parameters for the given training data. Among many options available to regularize the model, in the experiments, the value K_q is discounted by the average values of the query term feature in the corpus. The estimated probability distribution can be written as

$$P(d|R) \propto \prod_{i} \beta_i^{n(q_i,d)} \tag{5.21}$$

where $\{\beta_i | i = 1, \ldots, k\}$ are parameters learned from the training data.

In query-likelihood models the documents are ranked based on the query likelihood probability, $P(Q|M_d)$. The simple model based on query terms makes the query term independence assumption and estimates the query-likelihood probability as

$$P(Q|M_d) = \prod_i p(q_i|M_d)^{n(q_i,d)} \tag{5.22}$$

Smoothed language model is used to estimate the term-likelihood probability, $P(q_i|M_d)$. Ignoring the probabilistic interpretation of query-likelihood probability in (5.22) and maximum entropy model using query terms as features in (5.21), the similarity of the two formula can be observed as measure the relevance status value of a document for a given query. The β_i in (5.21) quantify the contribution of the query term q_i towards relevance and is estimated from a set of highly ranked documents; whereas the term-likelihood probability $P(q_i|M_d)$ is estimated from the empirical probability of the term in the document and its corpus probability.

5.6.2 Matched Query Terms Counts

In logistic regression method [17, 13], the number of matched query terms in a document is used as a feature in the logistic model. The query-likelihood and document-likelihood models discussed thus far are based on word occurrence statistics and it is difficult to incorporate number of query term matches in the likelihood models. In the whole document maximum entropy model, matched query term counts can be incorporated as a feature.

$$f_i(d) = f_Q(d) = \sum_{i=1}^{k} \delta(q_i, d) \tag{5.23}$$

where $\delta(q_i, d)$ is the indicator function for the occurrence of query term q_i in document d.

5.6.3 Query Term Co-occurrence Features

Co-occurrence of query terms is used in N-gram language models for information retrieval. They approximate the query-likelihood probabilities using N-gram statistics. In the context of whole document maximum entropy model, N-grams using the terms in the query are represented as constraints. The feature function for bigram is given by

$$f_{q_{i-1},q_i}(d) = n_{bg}(q_{i-1}, q_i, d) \tag{5.24}$$

where $n(q_{i-1}, q_i, d)$ is the number of occurrences of the bigram (q_{i-1}, q_i) in the document d. This feature constraints the bigram counts observed in relevant document, on the average, should match the empirical bigram count in the highly-relevant documents.

Biterm language models, proposed in Chapter 3, relaxed the word order constraint in bigram language models. These can be also be represented as constraints in estimating whole-document maximum entropy models. The corresponding feature function is given by

$$f_{q_{i-1},q_i}(d) = n_{bt}(q_{i-1}, q_i, d) \tag{5.25}$$

where $n(q_{i-1}, q_i, d)$ counts the occurrences of the word pair $\{q_{i-1}, q_i\}$ in the document d.

5.6.4 Local Context of Query Terms

Term co-occurrence features have been used in prior retrieval experiments. The local context analysis [82] and the relevance-based language model for retrieval [39] use term co-occurrence information to improve retrieval performance. The terms that co-occur with the query terms can be viewed to define the context of the query. One is interested in retrieving document with similar context. The feature function used as a constraint in the maximum entropy model is given by

$$f_{q,w}(d) = n(q, w, d) \tag{5.26}$$

where $n(q, w, d)$ is the number of co-occurrences of the query term q with a word w in the document d. Co-occurrence is based on the two terms appearing within a window of given number of words.

5.7 Experimental Results and Observations

The experiments were conducted on the Associated Press from TREC disks volume 1 and 2 (AP12), Wall Street Journal sub-corpus of TREC4 (WSJ2), TREC4 corpus and TREC8 corpus. TREC topics 101-150 are used for the AP12 collection, TREC4 queries (topics 201–250) were used for the WSJ2 and TREC4 corpora and the TREC8 queries (topics 401–450) were used against TREC8 collection. The retrieval performances using maximum entropy models are compared against the best performing smoothed unigram language models. The objective of these experiments is to demonstrate that the whole document maximum entropy model provides a framework for incorporating a variety of features towards estimating document likelihood probability.

Table 5.1 identifies the different combination of features that have been used in the experiments. The codes in the first column are used to identify the particular retrieval

Run	Description
ME-Q	Query term hits with expectation values discounted using corpus frequency of the query terms
ME-QBG	ME-Q + query bigram features
ME-QBT	ME-Q + query biterm features
ME-QC	ME-Q + local context of query terms with window size of 1.
ME-QN	ME-Q + matched query term counts
ME-QBTC	ME-Q + query bigram features + local context of query terms with window size of 1
ME-QBTN	ME-QBT + matched query term counts
ME-QCN	ME-QC + matched query term counts
ME-QBTCN	ME-QBTC + matched query term counts

Table 5.1: Combination of Features experimented with using Whole-Document Maximum Entropy Model

model in the results presented in this section. The different runs of maximum entropy models is compared with smoothed unigram language models (SUG-DIR). Dirichlet smoothing using prior value of $\mu = 1000$ is used. SUG-DIR is also used for identifying highly-relevant documents and sample documents for estimating the maximum entropy model. Top 100 documents from the results of smoothed unigram language model is used for the set of highly ranked documents and top 500 documents is used as a sample of computing the expectations of the different features using the estimated distribution.

Table 5.2 compares the retrieval performance of maximum entropy model using different feature combinations with smoothed unigram language models for the AP12 collection. The mean average precision, recall, R-precision and initial precision values are compared. The percentage change in these values over smoothed unigram model is given in the table. Using the same type of information or features, ME-Q improves average precision values by 2.51% over smoothed unigram models. In the case of AP12 collection, improvements in the retrieval performance is observed with each additional features. However, these changes are not statistically significant. Among the features considered, largest improvement is obtained by using the query terms and biterm features (ME-QBT). Bigram features perform equally for this query set. For this document collection and the set of query topics, other feature combinations have lesser contribution towards improving the retrieval performance.

Run	AvgP	% chg	Recall	% chg	RPrec	% chg	InitPr	% chg
SUG-DIR	0.2864	-	3576/4805	-	0.3177	-	0.7038	-
ME-Q	0.2936	2.51	3574/4805	-0.06*	**0.3205**	0.88	**0.7173**	1.92
ME-QBG	0.2953	3.11	3574/4805	-0.06*	0.3201	0.76	0.7126	1.25
ME-QBT	**0.2955**	3.18	3574/4805	-0.06*	0.3193	0.50	0.7126	1.25
ME-QC	0.2929	2.27	3575/4805	-0.03*	0.3139	-1.20	0.6896	-2.02
ME-QN	0.2941	2.69	3574/4805	-0.06*	0.3204	0.85	0.7165	1.80
ME-QBTC	0.2924	2.09	3575/4805	-0.03*	0.3144	-1.04	0.6751	-4.08
ME-QBTN	0.2928	2.23	3575/4805	-0.03*	0.3193	0.50	0.6979	-0.84
ME-QCN	0.2934	2.44	3575/4805	-0.03*	0.3162	-0.47	0.7028	-0.14
ME-QBTCN	0.2926	2.16	3575/4805	-0.03*	0.3154	-0.72	0.6665	-5.30

Table 5.2: Comparison of Smoothed Unigram Language Model and Whole-Document Maximum Entropy Models for AP12 collection

Table 5.3 compares the results of different maximum entropy model runs on the WSJ2 corpus. A different pattern is observed in terms of the contribution of the feature sets towards estimating relevance and information retrieval performance. The maximum entropy model using query term hits information alone (ME-Q) has statistically significant reduction in performance over smoothed unigram model. The star in the % change column indicates statistically significant difference as measured using one-sided Wilcoxon test at 0.05 level. The actual values are not included in the table for clarity of presentation. The table indicates that retrieval performance improves with addition of new features. The matched term counts feature that provided appreciable improvement on the AP12 collection (see Table 5.2) has no effect on the WSJ2 collection (ME-QN vs ME-Q). However, using this feature along with the local context of query terms (ME-QCN) provides the largest improvement (4.5%) for the WSJ2 corpus. Similar performance is obtained using bigram and biterm features.

Run	AvgP	% chg	Recall	% chg	RPrec	% chg	InitPr	% chg
SUG-DIR	0.2282	-	973/1183	-	0.2395	-	0.6124	-
ME-Q	0.2033	-10.91*	970/1183	-0.31	0.2230	-6.89*	0.4805	-21.54*
ME-QBG	0.2201	-3.55	970/1183	-0.31	0.2331	-2.67	0.5107	-16.61
ME-QBT	0.2203	-3.46	970/1183	-0.31	0.2331	-2.67	0.5217	-14.81
ME-QC	0.2290	0.35	971/1183	-0.21	0.2404	0.38	0.6233	1.78
ME-QN	0.2033	-10.91*	970/1183	-0.31	0.2230	-6.89*	0.4805	-21.54*
ME-QBTC	0.2321	1.71	971/1183	-0.21*	0.2414	0.79	0.6251	2.07
ME-QBTN	0.2296	0.61	970/1183	-0.31	0.2426	1.29	0.5576	-8.95
ME-QCN	**0.2384**	4.47	971/1183	-0.21	**0.2472**	3.22	**0.6320**	3.20
ME-QBTCN	0.2323	1.80	971/1183	-0.21	0.2414	0.79	0.6257	2.17

Table 5.3: Comparison of Smoothed Unigram Language Model and Whole-Document Maximum Entropy Models for WSJ2 collection using top 100 documents as highly relevant

The results in Table 5.3 for the WSJ2 collection used the top 100 documents as highly relevant document in the maximum entropy models. WSJ2 is a smallest of the document collections used in the experiments. The results of the runs using a smaller size for the highly relevant document set (top 10 documents) is given in Table 5.4. The set of top 500 retrieved documents for a given query is used as a sample in the estimation process. Comparable performance is obtained in this setting between smoothed unigram model and maximum entropy model using the query term constraints. This indicates that the selection of highly-relevant is important for the performance of the system. Similar observations are made for retrieval models using pseudo relevance feedback in language modeling approaches [39, 37]. Figure 5.2 gives the corresponding precision recall curve that reflects the comparable performance of SUG-DIR and ME-Q at different recall levels.

Run	AvgP	% chg	Recall	% chg	RPrec	% chg	InitPr	% chg
SUG-DIR	0.2282	-	973/1183	-	0.2395	-	0.6124	-
ME-Q	0.2236	-2.02	971/1183	-0.21	0.2314	-3.38	0.5481	-10.50
ME-QBG	0.2341	2.59	971/1183	-0.21*	0.2325	-2.92*	0.6097	-0.44
ME-QBT	0.2349	2.94	971/1183	-0.21*	0.2375	-0.84*	0.6121	-0.05
ME-QC	0.2396	5.00*	971/1183	-0.21*	0.2453	2.42*	0.6236	1.83*
ME-QN	0.2247	-1.53	971/1183	-0.21	0.2292	-4.30	0.5612	-8.36
ME-QBTC	**0.2484**	8.85*	972/1183	-0.10*	0.2558	6.81*	**0.6329**	3.35*
ME-QCN	0.2387	4.60	971/1183	-0.21*	0.2453	2.42*	0.6162	0.62
ME-QBTN	0.2158	-5.43	951/1183	-2.26	0.2342	-2.21	0.5607	-8.44
ME-QBTCN	0.2482	8.76*	972/1183	-0.10*	**0.2562**	6.97*	0.6326	3.30*

Table 5.4: Comparison of Smoothed Unigram Language Model and Whole-Document Maximum Entropy Models for WSJ2 collection using top 10 documents as highly-relevant

The best performance is obtained using the feature set of biterms and local context of the query terms (ME-QBTC). Among the additional features considered, significant improvements is obtained using the local context of query terms features (ME-QC). Some of the improvements are statistical significant as indicated in the table by a star in the % change column. Figures 5.3 and 5.4 compare the smoothed unigram model with maximum entropy model using two and three features, respectively. While using the top 10 documents for the highly-relevant document set improves the overall performance, the relative contribution of the individual features remains the same. The relative ordering of ME-QN < ME-QBG < ME-QBT < ME-QC is observed both tables 5.3 and 5.4.

Table 5.5 presents results using the TREC4 collection. TREC4 corpus is the largest collection used in the experiments and WSJ2 is a subset of this collection. The TREC4 and WSJ2 collections use the same query topics – TREC topics 201-250. In this

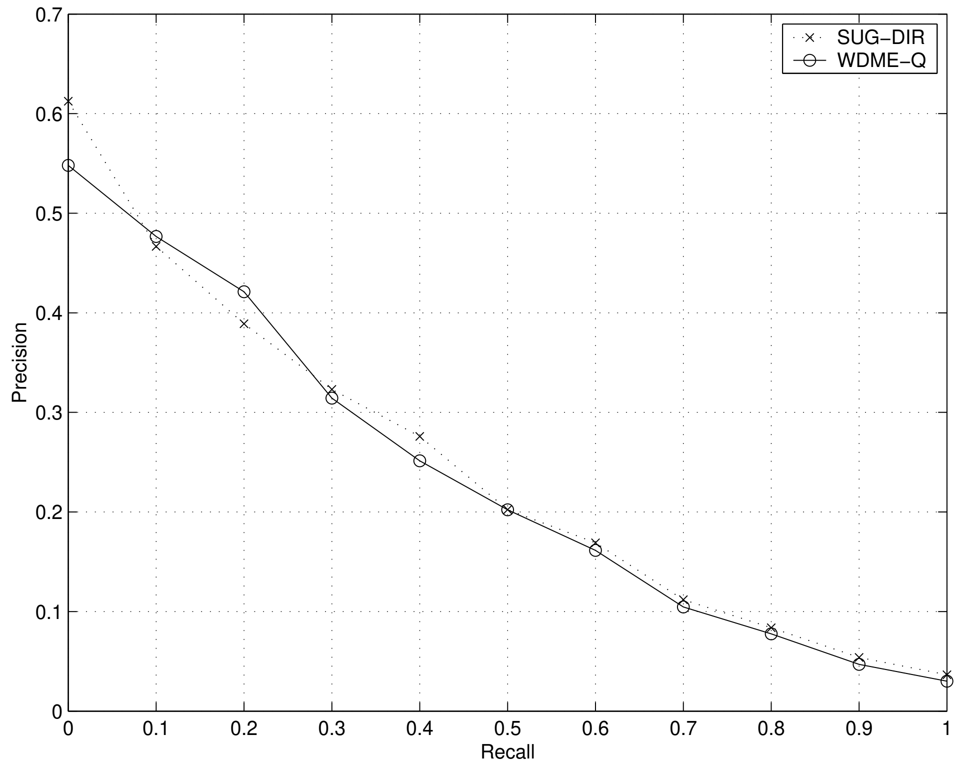

Figure 5.2: Precision-Recall Curve for WSJ2 collection – Smoothed Unigram and Whole Document Maximum Entropy Model using Query Term Constraints

case, however, the best performance is obtained using the query term and matched term count features with 3.57% improvement in average precision. While the difference is statistically significant, recall reduces significantly. It is noted that using the linguistic clues of bigrams and local context of query terms results in improvement of the average precision and R-precision values. With the same query topics used on WSJ2 and TREC4 collections, the effect of the corpus is observed in the relative ordering of the average precision values of the runs using two features. Using local context is better than bigrams in both collections (with the ordering of ME-QN $<$ ME-QW $<$ ME-QC from Table 5.3), however, matched term count performs better for TREC4 (ME-QBG $<$ ME-QBT $<$ ME-QC $<$ ME-QN). Further experiments are required to analyze the effect of corpus on feature selection.

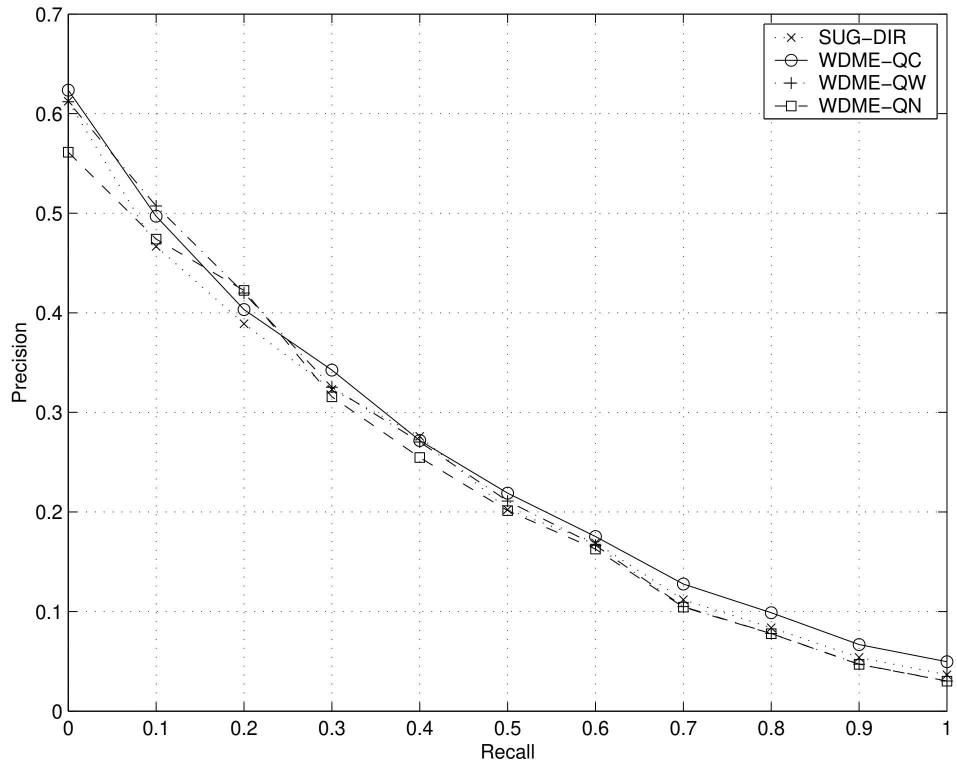

Figure 5.3: Precision-Recall Curve for WSJ2 collection – Smoothed Unigram and Whole Document Maximum Entropy Model using Two features

Comparison of performance of the smoothed unigram model and maximum entropy models on TREC8 collection is given in Table 5.6. In this collection, none of the feature sets explored in the experiments improved over the baseline smoothed unigram language model. It is noted that the concept language models, proposed in Chapter 4 that showed significant improvements in different document collections over smoothed unigram language models, improved the average precision value from 0.2610 to 0.2638. The difference is not statistically significant. This seems to be a characteristic of the query topics. Table 5.7 gives the performance results on the same document collection using TREC topics 351–400. The average precision values improve for some of the maximum entropy runs. However, the improvements are not statistically significant.

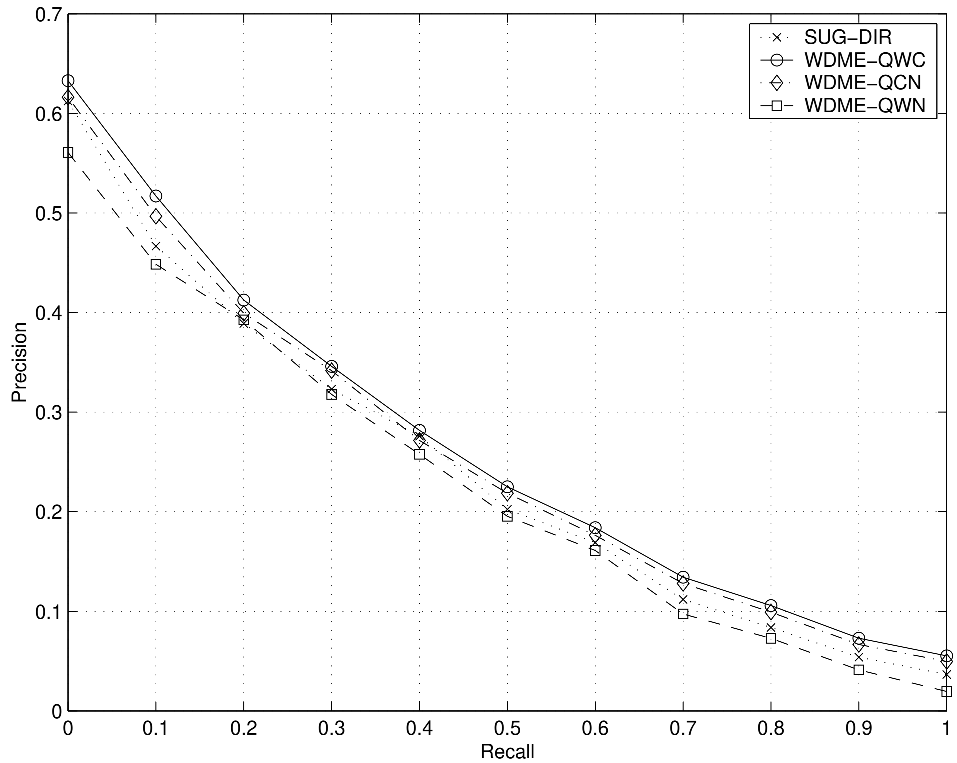

Figure 5.4: Precision-Recall Curve for WSJ2 collection – Smoothed Unigram and Whole Document Maximum Entropy Model using Three features

Figure 5.5 compares the average precision values for all retrieval models and all document collections. Across collections, the combination of biterm and local context of query terms seems to positively effect performance. Mixed results are obtained using the matched query terms counts. Figure 5.6 compares R-precision values across different document collections and features. The contribution of the matched query term counts is observed in improvement in R-precision values for all except AP12 and TREC8 collections.

Table 5.8 compares the different language models proposed in this dissertation to the smoothed unigram language models. In the case of whole document maximum entropy models, the metrics correspond to the feature set with the best average-precision value. The concept language models perform very well across document collections. Concept un-

Run	AvgP	% chg	Recall	% chg	RPrec	% chg	InitPr	% chg
SUG-DIR	0.1876	-	3276/6503	-	0.2475	-	**0.6622**	-
ME-Q	0.1841	-1.87	3272/6503	-0.12	0.2416	-2.38	0.6052	-8.61
ME-QBG	0.1873	-0.16	3273/6501	-0.09	0.2436	-1.58	0.6382	-3.62
ME-QBT	0.1878	0.11	3273/6501	-0.09	0.2436	-1.58	0.6384	-3.59
ME-QC	0.1917	2.19	3276/6503	0.00	0.2505	1.21	0.5975	-9.77
ME-QN	**0.1943**	3.57*	3064/5918	-6.47*	0.2489	0.57*	0.6413	-3.16*
ME-QBTC	0.1930	2.88	3276/6501	0.00	**0.2509**	1.37	0.6102	-7.85
ME-QBTN	0.1926	2.67	3273/6501	-0.09	0.2473	-0.08	0.6459	-2.46
ME-QCN	0.1878	0.11	3276/6503	0.00	0.2461	-0.57	0.5855	-11.58
ME-QBTCN	0.1929	2.83	3276/6501	0.00*	0.2508	1.33	0.6092	-8.00

Table 5.5: Comparison of Smoothed Unigram Language Model and Whole-Document Maximum Entropy Models for TREC4 collection

igram language model estimate the concept likelihood probability using a bigram approximation of the sequence of words that constitute the concept. While concept occurrence can be used as a feature and corresponding constraint can be used in the maximum entropy model, identifying and counting concept occurrences requires additional processing of the document corpus. Approximating concepts through their bigram or biterm approximations corresponds to using the bigrams or biterms as features in the maximum entropy model. While selection of feature set is an issue that needs to be addressed in the context of whole-document language models, they improve over the performance of bigram and biterm language models in all except the TREC8 collection.

The following observations are in order based on the experiments on the maximum entropy models:

- Comparable performance using query term constraints with that of smoothed unigram language models verify the applicability of the sampling method used for com-

Run	AvgP	% chg	Recall	% chg	RPrec	% chg	InitPr	% chg
SUG-DIR	**0.2610**	-	2970/4728	-	**0.3040**	-	**0.7623**	-
ME-Q	0.2473	-5.25	2969/4728	-0.03	0.2959	-2.66*	0.7188	-5.71*
ME-QBG	0.2500	-4.21	2969/4728	-0.03	0.2981	-1.94*	0.7116	-6.65*
ME-QBT	0.2496	-4.37	2969/4728	-0.03	0.2948	-3.03*	0.7112	-6.70*
ME-QC	0.2388	-8.51*	2970/4728	0.00*	0.2929	-3.65*	0.7237	-5.06*
ME-QN	0.2485	-4.79	2969/4728	-0.03	0.2948	-3.03*	0.7318	-4.00*
ME-QBTC	0.2405	-7.85*	2970/4728	0.00*	0.2930	-3.62*	0.7321	-3.96*
ME-QBTN	0.2494	-4.44	2970/4728	0.00	0.3000	-1.32*	0.7101	-6.85*
ME-QCN	0.2389	-8.47*	2970/4728	0.00*	0.2929	-3.65*	0.7237	-5.06*
ME-QBTCN	0.2418	-7.36*	2970/4728	0.00*	0.2926	-3.75*	0.7308	-4.13*

Table 5.6: Comparison of Smoothed Unigram Language Model and Whole-Document Maximum Entropy Models for TREC8 collection using topics 401-450

Run	AvgP	% chg	Recall	% chg	RPrec	% chg	InitPr	% chg
SUG-DIR	0.1824	-	1710/3026	-	0.2451	-	**0.7062**	-
ME-Q	0.1820	-0.22	1706/3026	-0.23	0.2449	-0.08	0.6625	-6.19
ME-QBT	**0.1856**	1.75	1706/3026	-0.23	0.2457	0.24	0.7004	-0.82
ME-QC	0.1813	-0.60	1704/3026	-0.35	0.2486	1.43	0.6694	-5.21
ME-QN	0.1820	-0.22	1706/3026	-0.23	0.2449	-0.08	0.6625	-6.19
ME-QBTC	*0.1833*	0.49	1705/3026	-0.29	**0.2492**	1.67	0.6821	-3.41
ME-QBTN	*0.1842*	0.99	1706/3026	-0.23	0.2447	-0.16	0.6878	-2.61
ME-QCN	0.1814	-0.55	1704/3026	-0.35	0.2486	1.43	0.6696	-5.18
ME-QBTCN	0.1807	-0.93	1705/3026	-0.29	0.2491	1.63	0.6636	-6.03

Table 5.7: Comparison of Smoothed Unigram Language Model and Whole-Document Maximum Entropy Models for TREC8 collection using topics 351–400

puting the expectation values of the estimated probability distribution as well as the regularization/smoothing options employed.

- As with other retrieval experiments [39, 37] using highly-relevant documents to improve retrieval results, the different models proposed in this chapter depend on the size of the relevant set. However, this does not seem to have an effect on the contribution one can expect from the various features experimented with here. The relative ordering of performance improvements using the feature sets as observed in the case

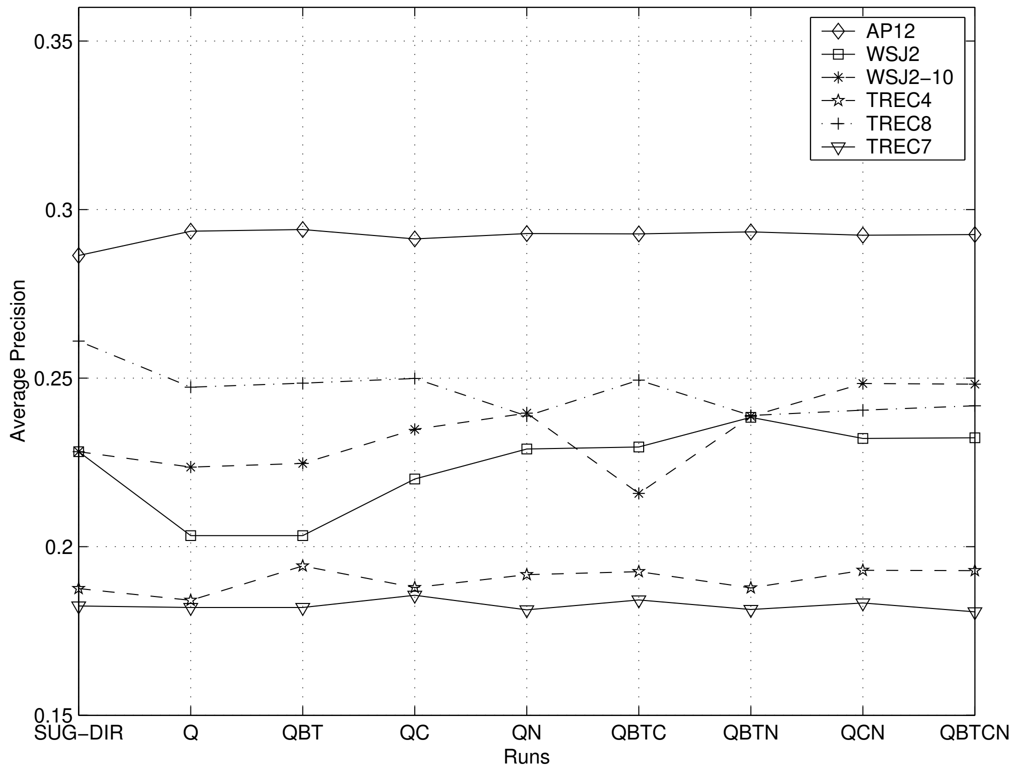

Figure 5.5: Average Precision Comparison across different collections using Smoothed Unigram and Whole Document Maximum Entropy Models

of experiments on the WSJ2 collection using different number of documents in the highly-relevant document subsets.

- The whole document maximum entropy model is an elegant way to incorporate non-textual clues like matched query term counts. In the experiments, its contribution is observed in improved retrieval performance by itself or in combination with other features.

Next section identifies a particular IR problem where the maximum entropy model can be applied to represent user's information need expressed as query text and metadata in terms textual and non-textual features.

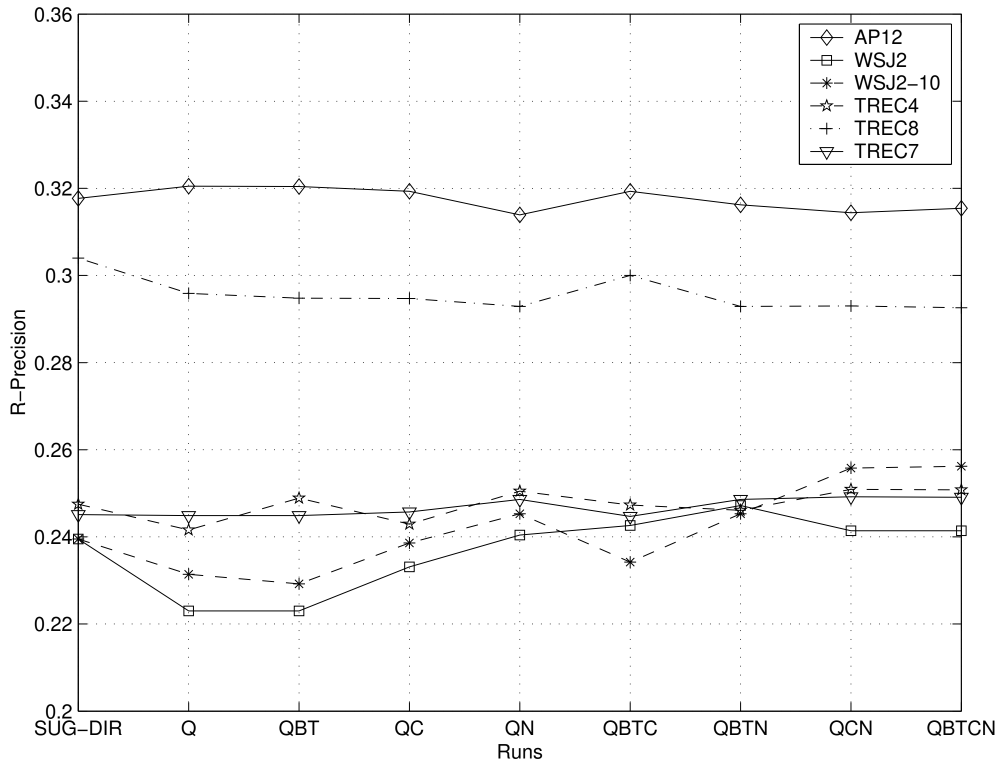

Figure 5.6: R-Precision Comparison across different collections using Smoothed Unigram and Whole Document Maximum Entropy Models

5.8 TREC HARD Task

Adhoc document retrieval focuses on topical relevance of a document to a given textual query. Extending beyond topical relevance, queries in TREC High Accuracy Retrieval of Documents (HARD) track characterize user's information need in the form of metadata for the query text. The HARD task is presented here as an example where the whole-document maximum entropy language models can be effectively used to rank documents to satisfy the diverse expression of user's information need. The HARD task is briefly described followed by experimental results using simple models for the metadata. The maximum entropy model can be used to solve the HARD problem by modeling the metadata requirements as constraints that the document or passage likelihood probability

Run	AvgP	% chg	Recall	% chg	RPrec	% chg	InitPr	% chg
WSJ2								
SUG-DIR	0.2282	-	973/1183	-	0.2395	-	0.6124	-
BG-DIR	0.2442	7.01	961/1183	-1.23	0.2475	3.34	0.6005	-1.94
BT-DIR	0.2440	6.92	965/1183	-0.82	0.2468	3.05	0.6227	1.68
CLM	0.2541	11.35	970/1183	-0.31*	0.2517	5.09*	0.6515	6.38*
ME-QBTC	0.2484	8.85*	972/1183	-0.10*	0.2558	6.81*	0.6329	3.35*
TREC4								
SUG-DIR	0.1876	-	3276/6503	-	0.2475	-	0.6622	-
BG-DIR	0.1938	3.30	3289/6501	0.40	0.2516	1.66	0.6631	0.14
BT-DIR	0.1929	2.83	3288/6501	0.37	0.2490	0.61	0.6918	4.47
CLM	0.2020	7.68	3345/6503	2.11	0.2589	4.61	0.6627	0.08*
ME-QN	0.1943	3.57*	3064/5918	-6.47*	0.2489	0.57*	0.6413	-3.16*
TREC8								
SUG-DIR	0.2610	-	2970/4728	-	0.3040	-	0.7623	-
BT-DIR	0.2480	-4.98*	2840/4728	-4.38*	0.2954	-2.83*	0.7516	-1.40*
BG-DIR	0.2538	-2.76	2900/4728	-2.36*	0.3028	-0.39	0.7393	-3.02
CLM	0.2638	1.07	2964/4728	-0.20	0.3114	2.43	0.8022	5.23
ME-QBG	0.2500	-4.21	2969/4728	-0.03	0.2981	-1.94*	0.7116	-6.65*

Table 5.8: Comparison of smoothed N-gram language models with biterm language models, concept language models and whole-document maximum entropy models for WSJ2, TREC4 and TREC8 corpus

should satisfy. The HARD corpus of 372,219 documents of 1.7Gb contains government documents and news articles.

Along with the query text, fixed-valued metadata is provided for each topic. The metadata includes purpose, genre and granularity expected for the query. Purpose indicates why the user is searching for the information and takes BACKGROUND, DETAILS, ANSWER or ANY as its values. Genre corresponds to the type of material of interest to the searcher and its takes OVERVIEW (general news on topic), REACTION (news commentary), I-REACTION (reaction from international news commentary), ADMINISTRATIVE (US government sources) or ANY. Granularity specifies the amount of text the search anticipates for the query. Its value ranges from DOCUMENT, PASSAGE, SENTENCE, PHRASE or ANY. In addition to satisfying the query, the retrieved documents or passages should satisfy the metadata value specified by the user. Topical relevance is identified as soft-relevance to given query in the HARD track and Hard relevance corresponds to satisfying the query text and metadata requirements of the query. Relevance was also measured at the document level and passage level. The passage level relevance corresponds to satisfying the GRANULARITY metadata of the user query.

For HARD retrieval, some model of the metadata is required to identify documents that satisfy the metadata values and weight their relevance to the given query. For a given query, text snippets are short-listed based on their satisfying the user's query and granularity requirements. A number of keyword- and non-keyword based features are used to weight the snippets and rank them. A text snippet is viewed as a sequence of keyword and non-keyword features and a model is associated with it in the language modeling sense. Unlike a whole-document maximum entropy model that views a document or text snippet as a "bag of features" and capture the depedencies of the features using the maximum entropy model, independence of the features used is assumed here. Text snippets are weighted based on the probability of generating a given feature. Smoothed probabilities are estimated for keyword features and empirical probabilities are used for non-keyword features. Ad-hoc query weights are assigned for these features based on the metadata values.

Documents and queries are processed through Cymfony's Information Extraction tool – InfoXtract [1]. Genre metadata is handled as follows: *Reaction* or *I-Reaction* typically

[1] www.cymfony.com

involve entities grouped as *actors*. This includes persons, organizations, and government entities. The occurrence of such entities triggers Reaction or I-Reaction genre type. If the origin of such an actor is a US location, the text snippet is most likely to be a Reaction than I-Reaction. InfoXtract engine tags associations or relations that can link entities of one type with another. To determine if the text snippet is Reaction or I-Reaction, entities associated with *actors* are searched to see if they include location entities. Location names are checked against a lexicon of US cities, counties and states to determine if the origin of the *actor*. A number of US cities share names with non-US locations - e.g. *Moscow* and *China*. In such instances, the context of the location name is checked. If the next word/phrase is a US location or the phrase "United States" or its variations, the location is classified as a US location. Document-source based features are used to weight text snippets against genre value of *Administration*. We did not eliminate all non-government sourced documents but assigned lower weights than government sourced documents.

The location of the text snippet in a document, presence or absence of *details* (described below) information and the query term coverage were used as triggers for the different values of purpose metadata. It is assumed that a text snippet provides details on a particular topic of interest if it contains numeric, percentage, frequency and time information. Such information, extracted by InfoXtract, are grouped together as *details* information and snippets are weighted based on the frequency of such information. The absence of such information is also used as a feature. The snippet location feature gives more weights to snippets closer to the center of the document. It is based on the heuristic that details and background information are typically found in the middle rather than at the start or end of the document. Queries with metadata background or details assign more weights to this feature than queries with metadata or answer or any.

Section 4.5.5 presented the result of adhoc retrieval using the smoothed unigram and concept language models. Table 5.9 shows the average and R-precision values for the different runs along with the average number of relevant documents among the top 10 retrieved documents. SUGT-DIR and SUGTD-DIR correspond to smoothed unigram language models using title only and title and description, respectively, as queries. Statistically significant improvement is obtained using concept language models (CLMTD-DIR) with title and description queries. CLMTD-FS corresponds to document retrieval results that satisfy the query text and meta data using the methods given above. The best, median and worst performance measure based on runs submitted for this task is given for comparison.

	Rel. Ret. @10	Avg. Prec.	R-Prec.
SUGT-DIR	5.375	0.3126	0.3335
SUGTD-DIR	5.750	0.3396	0.3719
CLMTD-DIR	5.854	0.3540	0.3784
CLMTD-FS	3.583	0.2073	0.2562
Best	6.5	0.4069	0.4250
Median	4.729	0.2841	0.2994
Worst	0.417	0.0026	0.0038

Table 5.9: TREC HARD task - Soft Document Relevance Comparison

Modest improvement was achieved under hard-relevance condition as given Table 5.10. Among other things, the selection of all possible relevant snippets in a document deemed highly-relevant as compared to selecting only 2-3 snippets from a document is a possible reason for the significant drop in performance for CLMTD-FS.

	Rel. Ret. @10	Avg. Prec.	R-Prec.
SUGT-DIR	4.042	0.2543	0.2764
SUGTD-DIR	4.521	0.2904	0.3230
CLMTD-DIR	4.542	0.2981	0.3286
CLMTD-FS	2.771	0.1726	0.2078
Best	5.271	0.3875	0.3604
Median	3.792	0.2673	0.2490
Worst	0.312	0.0038	0.0024

Table 5.10: TREC HARD task - Hard Document Relevance Comparison

	Psg. Prec. @10	R-Prec.	F @ 30
SUGT-DIR	0.2399	0.1594	0.1178
CLMTD-DIR	0.2987	0.1997	0.1528
CLMTD-FS	0.2313	0.1699	0.0798
Best	0.3973	0.3195	0.1738
Median	0.2574	0.1794	0.1000
Worst	0.0136	0.0046	0.0010

Table 5.11: Passage Relevance Comparison

Table 5.11 gives the performance measures for passage relevance. The first three runs were evaluated with the whole document being the retrieved passage. CLMTD-FS included passage level results corresponding to the granularity metadata. Adhoc weighting scheme was used to weight the features. A maximum entropy model that can learn the weights of these features from highly relevant documents can result in better results.

In HARD, relevance of a document to a given query is not only based on satisfying the query text but also the associated metadata. This yields well to the generation of a relevance or query model for the given information need that captures both textual and non-textual characteristics. The metadata value of ADMINISTRATIVE for genre can be implied from the source of the document – whether it is a government document or a new article. Such information can easily placed as an expected attribute or feature in the relevance model and its weights learned in the maximum entropy model. Additional modeling and retrieval experiments are required to establish the applicability of maximum entropy model to the HARD task. However, the HARD corpus yields well for the application of the whole-document maximum entropy model and is identified as a direction of future work.

5.9 Summary

A retrieval method that incorporates diverse information sources of relevance using maximum entropy framework is proposed in this chapter. Similar to the whole sentence maximum entropy model that was proposed for estimating sentence likelihood in speech recognition, the whole document maximum entropy model estimates directly the document likelihood probability. A document is modeled as a “bag of features”. Highly-relevant documents for a given query are used to define constraints for estimating document likelihood probability. The constraints represent features that contribute towards estimating relevance to the given query. The results presented in this chapter used query term hits, matched term counts, bigrams and local context of query terms as features. Non-textual features like matched query term counts are difficult to incorporate in document likelihood models based on chain-rule of probabilities. The maximum entropy approach provides an elegant approach to incorporate diverse features.

The whole document maximum entropy model using query term hits as features gives comparable performance to that of smoothed unigram language models. Additional features provide improvements in three of the four document collections. The performance of the model depends on the features, queries and corpus information. Some assumptions have been used in estimating the document likelihood probability as a maximum entropy model. The feature weights are learned from a training document set of highly-relevant documents. Top 10 or 100 documents were used to identify the set of highly-relevant documents. The performance of the retrieval model depends on this threshold and similar observations have been made for other retrieval models that use the top N documents as highly-relevant to improve retrieval. The amount of information a feature contributes towards relevance seems to be independent of this threshold as observed in the WSJ2 collection. However, more experiments are required to explore the relation between the features and the selection of highly-relevant documents. Relevance provided by the user in an interactive system would help in.

The whole document maximum entropy problem has two computationally intractable sub-problems of computing the expectation over all documents and the normalization constant. While the normalization constant is not required for relative ordering of documents, the expectation computation is handled using sampling. Instead of generating sample, the top M documents are selected to represent the sample. The threshold M is assumed to be larger than the threshold N used for identifying highly-relevant documents. Top 500 documents were used in the experiments. Larger sized samples would provide better estimates but not necessarily lead to significant retrieval performance improvements.

With the probability estimation problem addressed through the use of maximum entropy principle, the whole document maximum entropy model can be viewed as a method that "reduces" the retrieval problem to that of feature selection. Selecting the best set of features to characterize the query should result in significant improvements in retrieval performance.

The results presented in this chapter can be characterized as initial effort in exploring the retrieval model that formalizes the probability estimation effort for retrieval and incorporates diverse set of features towards estimating relevance for a given query. Better methods for identifying and selecting features need to be explored. The initial results indicate some correlation between features and the document collection. Methods to explore or formalize the correlation will help in building better retrieval models. The maximum entropy optimization solution can be improved using better estimation and sampling techniques.

Prior queries to a particular document collection have clues that can be effectively used in maximum entropy framework. The experiments and the features discussed in this chapter are based on learning from highly-relevant documents. Features that cut across queries like matched query term counts have to be explored. In the maximum entropy framework, weights for such features can be learned from prior queries and relevance judgments.

Chapter 6

Conclusion and Future Work

The chapter presents a summary of the research contributions of this dissertation and identifies directions for future research.

6.1 Summary

The following, in the context of language modeling approaches for information retrieval that incorporate additional features beyond term occurrences, were identified as the central questions of this dissertation (ref. Chapter 1):

Q1 "Have these techniques failed to work because the necessarily smooth retrieval functions have not allowed them to?"

Q2 "Have they failed to work because the information they provide is overcome by estimation error?" or,

Q3 "Have they failed to work because there is no new information to be gained from them?"

This dissertation addressed and provided some answers to these questions.

New approaches for incorporating query features beyond term occurrences in the language modeling approach to information retrieval were proposed in this dissertation and empirically shown that such features do contribute towards improved retrieval performance. However, it was identified that the features should be selected to reflect the particular requirements of the application. In other words, language models have to be *designed* for information retrieval. The biterm features introduced here are similar to bigrams, except that the word order is relaxed. The word order constraint is a requirement for speech recognition and N-gram models performed well in those settings. Relaxing that constraint for information retrieval resulted in improved average precision values.

It was identified that relaxing the word order constraint fails to capture some of the concepts expressed in queries. Viewing bigrams as a pair of trigger and triggered term, the Maximum bigram language model estimates term probabilities based on the best "triggers" for the query terms in a document. The triggers are selected on a document-by-document basis. Prior work concentrated on exploring different approaches and frameworks for solving the information retrieval problem using statistical language models and explored features that worked well in speech recognition. This is the first work which identified the need to select features based on their relevance to the application of interest.

Concept language models was proposed as a family of language models that take a different approach to modeling the querying process. A user query is viewed as a sequence of concepts that the user is interested in and the query text is an expression of these concepts of interest. Syntactic parse of the user query is used to generate concept language models and documents are ranked based on the likelihood of their models generating the sequence of concepts. Different approximations for estimating the concept and query-likelihood probabilities were presented. While syntactic parsing of documents and queries have been explored in prior IR experiments, this is the first effort to use them in the settings of language modeling approaches and the experimental results demonstrate that such a representation results in performance improvements that are statistically significant for some datasets.

The biterm, maximum bigram and concept language models proposed in this dissertation answer the questions Q2 and Q3 in the negative. Language features beyond term occurrences do provide information that can be utilized effectively in language models *designed* for information retrieval and that estimation errors can be overcome by using alternate representations for queries.

The smooth retrieval functions used in the language modeling approaches are based on the chain rule of probabilities. The query-likelihood probability is estimated as the joint probability of the sequence or set for query terms using the chain rule. While different approximations for the term-likelihood probabilities have been explored, the implicit sequencing of terms in the estimation restricts other features to be elegantly included in retrieval function. The maximum entropy language model approach is shown to facilitate diverse information sources of relevance to be incorporated in the retrieval function and results in improved or comparable retrieval performance. A document is represented as a "bag of features" and the document likelihood probability is estimated directly as a whole document maximum entropy model. Textual (query term hits, bigrams, local context of query terms) and non-textual (matched query term counts) were used as features and experiments on standard document collections indicate that the whole document maximum entropy model indeed is a mechanism to include additional features in information retrieval and highlights the contribution of the additional features.

6.2 Future Research Opportunities

A number of factors contribute to the relevance of a document to a user's information need. Term occurrence statistics have been widely used in different retrieval models. But matching terms between query and document is one particular aspect of satisfying user's information need. Methods are required to capture, represent and incorporate the other relevance related factors to improve retrieval performance. Some of the possible extensions to the models proposed in this dissertation along with applications where such extensions would help is given below.

Identification of IR-specific features: The most interesting aspect of language modeling approach to information retrieval is that language models have the flexibility to capture any computable feature and enable inference based on them. While initial interest in language modeling approach to information retrieval has been in defining new models and approaches, better representations in terms of N-gram models have been explored. Representations that capture the specific requirements of IR are essential for improving retrieval performance.

Word order constraint: Relaxing the word-order constraint resulted in improvements in retrieval performance. It was noted that the word-order constraint cannot be arbitrarily relaxed. Language models that can identify when to relax and when to enforce such word ordering constraints will result in better representation of user queries and possibly lead to better retrieval results.

Concept Identification and Representation: In the concept language models, contiguous words were identified as concepts. While this works well in most settings, concepts that can be inferred from, say, two noun phrases, can also be represented as a concept in the concept language models. The two-layer models enables some structure information to be encoded in the model and this can be used during retrieval. While the concept language models used only the parse tree of the query, the output of the information extraction system contained additional information like named entities, relations. Such information do provide more insights into the information need of the user. Language models have to be designed to incorporate such information and that their design should be in such a way that it effects better information retrieval. It was also identified that the concept language models provide an elegant method to incorporate concept synonyms in the language modeling framework. This is a possible future direction of work on concept language models.

Information sources for Relevance: The language modeling approach to information retrieval has been described to have "reduced" the information retrieval problem to that of probability estimation and smoothing. By assuming a particular framework – maximum entropy principle, here – the whole document maximum entropy method proposed in Chapter 5 provides a unified framework for the estimation process and can be described to have "reduced" the information retrieval to that of selecting information sources of relevance for a given query. A number of keyword based features have been explored in prior IR work in vector space and probabilistic models. Statistical variations of term frequency and inverse document frequency have been explored in logistic regression models. Syntactic and Semantic clues in queries and documents are information sources of relevance for given query. Further work is required to identify such clues and incorporate them in the retrieval framework.

User modeling and search personalization: Information retrieval has always been explored with the assumption of documents are either relevant or non-relevant to a given topic and independent of specific user requirements. Search personalization based on user profiles is being realized through adhoc techniques of search followed by result filtering or reweighting. With the ability to incorporate diverse information sources for relevance, that includes user's preference of a particular topic or sense of a word, the whole-document maximum entropy model can be explored as a possible approach to search personalization. The performance of maximum entropy models depends on the relevance of the features selected in the model. Some interesting questions in this context are: What features can be captured as constraints in the maximum entropy model? How to evaluation and select features that contribute to the relevance of the user query that will lead to satisfying user's information need?

Other text processing applications: The information retrieval problem of adhoc document retrieval was addressed in this dissertation. Some of the proposed models can be explored for related problems in other text processing/information retrieval applications like text categorization, filtering. Maximum entropy models have been explored for text categorization. The idea of relevance models or query models yields well to text categorization. A model, C, using the training documents, can be generated for each category, c. This represents the desired characteristics of an "ideal" document in that category. The likelihood of a document with those features measured by the probability $P(d|C)$ can be used to classify the document to be in or not-in the category. Limited experiments have been performed on the use of information beyond term occurrence in text categorization. With relevance feedback from users in the case of text filtering, it is possible to build a topic or relevance model for the topic of interest and the likelihood of the incoming document being generated by that topic model can be used for making the filtering or routing decision. It will be interesting to see if the concept language models and whole-document maximum entropy models proposed here will work in the context of text categorization and filtering applications.

Bibliography

[1] J. Allan. HARD Track Overview in TREC 2003: High Accuracy Retrieval from Documents. In *Proceedings of TREC-12*, Gaithersburg, CA, 2003. NIST.

[2] J. Allan and H. Raghavan. Using Part-of-Speech Patterns to Reduce Query Ambiguity. In *Proceedings of SIGIR*, 2002.

[3] L. R. Bahl, F. Jelinek, and R. L. Mercer. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:179–190, 1983.

[4] D. Beeferman, A. Berger, and J. Lafferty. Statistical Models for Text Segmentation. *Machine Learning*, 34:1–34, 1999.

[5] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. Bridging the Lexical Chasm: Statistical Approaches to Answer-Finding. In *Proceedings of SIGIR*, pages 192–199. ACM, New York, 2000.

[6] A. Berger and J. Lafferty. Information Retrieval as Statistical Translation. In *Proceedings of SIGIR'99*, pages 222–229, 1999.

[7] A. Berger and V. Mittal. OCELOT: a system for summarizing web pages. In *Proceedings of SIGIR*, pages 144–151. ACM, New York, 2000.

[8] A. Bookstein and D. Swanson. Probabilistic Models for Automatic Indexingx. *Journal of the Americal Society for Information Science*, 25(5):312–318, 1974.

[9] C. Cai, R. Rosenfeld, and L. Wasserman. Exponential Language Models, Logistic Regression, and Semantic Coherence. In *In Proc. NIST/DARPA Speech Transcription Workshop*, 2000.

[10] J. P. Callan, W. B. Croft, and S. M. Harding. The INQUERY Retrieval System. In *Proceedings of the Third International Conference on Database and Expert Systems Applications*, pages 78–83, 1992.

[11] M. F. Caropreso, S. Matwin, and F. Sebastiani. Statistical phrases in automated text categorization. Technical Report IEI-B4-07-2000, Istituto di Elaborazione dell'Informazione, Pisa, IT, 2000.

[12] C. Chelba and F. Jelinek. Exploiting Syntactic Structures for Language Modeling. In *Proceedings of the COLING-ACL Meeting*, pages 225–231, Montreal, Canada, 1998.

[13] A. Chen. Cross-language retrieval experiments at clef 2002. In *Proceedings of CLEF 2002*, 2002.

[14] S. Chen and R. Rosenfeld. Efficient Sampling and Feature Selection in Whole Sentence Maximum Entropy Language Models. In *In Proc.ICASSP '99*, 1999.

[15] S. F. Chen and R. Rosenfeld. A Survey of Smoothing Techniques for ME Models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50, 2000.

[16] W. S. Cooper. Exploiting the maximum entropy principle to increase retrieval effectiveness. *Journal of the Americal Society for Information Science*, 34(1):31–39, 1983.

[17] W. S. Cooper, F. C. Gey, and D. P. Dabney. Probabilistic Retrieval based on Staged Logistic Regression. In *Proceedings of SIGIR*, 1992.

[18] 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), 1998.

[19] W. B. Croft and Harper D. J. Using Probabilistic Models of Document Retrieval without Relevance Information. *Journal of Documentation*, 37:285–295, 1979.

[20] J. N. Darroch and D. Ratcliff. Generalized Iterative Scaling for Log-Linear Models. *Annals Math. Stats.*, 43:1470–1478, 1972.

[21] S. Deerwester, S. Dumais, T. Laudauer, G. Furnas, and R. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[22] V. Della Pietra, S.and Della Pietra and J. Lafferty. Inducing Features of Random Fields. *IEEE trans. on PAMI*, 19(4):380–393, 1997.

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[24] N. Fuhr. Models for Retrieval with Probabilistic Indexing. *Information Processing and Management*, 25(1):55–72, 1989.

[25] N. Fuhr and C. Buckley. A Probabilistic Learning Approach for Document Indexing. *ACM Transactions on Information Systems*, 9(3):223–248, 1991.

[26] N. Fuhr and C. Buckley. Optimizing Document Indexing and Search Term Weighting bsed on Probabilistic Models. In *Proceedings of TREC-1*, Gaithersburg, CA, 1993. NIST.

[27] N. Fuhr and U. Pfeifer. Combining Model-Oriented and Description-Oriented Approaches for Probabilistic Indexing. In *Proceedings of SIGIR*, 1991.

[28] E. Greengrass. Information Retrieval: A Survey.

[29] W. R. Greiff and J. M. Ponte. The Maximum Entropy Principle and Probabilistic IR Models. *ACM Transactions on Information Systems*, 18(3):246–287, 2000.

[30] D. Hiemstra. A Linguistically Motivated Probabilistic Model of Information Retrieval. In *European Conference on Digital Libraries*, pages 569–584, 1998.

[31] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, 2001.

[32] E.T. Jaynes. Information Theory and Statistical Mechanics. *Physical Reviews*, 106:361–373, 1957.

[33] P. B. Kantor and J. J. Lee. The Maximum Entropy Principle in Information Retrieval. In *Proceedings of SIGIR*, 1986.

[34] P. B. Kantor and J. J. Lee. Testing the Maximum Entropy Principle for Information Retrieval. *Journal of the American Society for Information Science*, 49:557–566, 1998.

[35] 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*, 35:400–401, 1987.

[36] S. Khudanpur and J. Wu. Maximum Entropy Techniques for Exploiting Syntactic, Semantic and Collocational Dependencies in Language Modeling. *Computer Speech and Language*, pages 355–372, 2000.

[37] J. Lafferty and C. Zhai. Document Language Models, Query Models, and Risk Minimization for Information Retrieval. In *Proceedings of SIGIR'01*, pages 111–119, 2001.

[38] J. Lafferty and C. Zhai. Probabilistic IR Models based on Document and Query Generation. In *Proceedings of the Workshop on Language Modeling and Information Retrieval*, 2001.

[39] V. Lavrenko and W. B. Croft. Relevance-based Language Models. In *Proceedings of SIGIR'01*, pages 120–127. ACM, New York, 2001.

[40] J. J. Lee and P. B. Kantor. A Study of Probabilistic Information Retrieval in the Case of Inconsistent Expert Judgments. *Journal of the American Society for Information Science*, 42:166–172, 1991.

[41] R. Malouf. A Comparison of algorithms for Maximum Entropy Parameter Estimation. In *Sixth Workshop on Computational Language Learning(CoNLL)*, 2002.

[42] E. L. Margulis. Modeling Documents with Multiple Poisson Distributions. *Information Processing and Management*, 29(2):215–227, 1993.

[43] M. E. Maron and Kuhns J. L. On Relevance, Probabilistic Indexing and Retrieval. *J. ACM*, 7:216–244, 1960.

[44] A. McCallum and K. Nigam. A Comparison of Event Models for Naive Bayes Text Classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[45] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E Teller. Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[46] D. R. H. Miller, T. Leek, and R. M. Schwartz. A Hidden Markov Model Information Retrieval System. In *Proceedings of SIGIR'99*, pages 214–221, 1999.

[47] M. Mitra, C. Buckley, A. Singhal, and C. Cardie. An Analysis of Statistical and Syntactic Phrases. In *Proceedings of RIAO-97, 5th International Conference "Recherche d'Information Assistee par Ordinateur"*, pages 200–214, Montreal, CA, 1997.

[48] R. Nallapati and J. Allan. Capturing Term Dependencies using a Sentence Tree based Language Model. In *Proceedings of CIKM'02*, pages 383–390, 2002.

[49] M. Narita and Y. Ogawa. The Use of Phrases from Query Texts in Information Retrieval. In *Proceedings of SIGIR'00*, 2000.

[50] National Institute of Standards and Technology (NIST). *Text REtrieval Conferences (TREC)*, Gaithersburg, MD., 1993-. URL: `http://trec.nist.gov`.

[51] National Library of Medicine. MEDLINE. available at `http://www.ncbi.nlm.nih.gov`.

[52] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[53] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 1999.

[54] F. Peng, D. Schuurmans, and S. Wang. Augumenting Naive Bayes Text Classifier with Statistical Language Models. *Information Retrieval*, pages 317–345, 2004.

[55] J. Ponte. Is Information Retrieval anything more than Smoothing. In *Proceedings of the Workshop on Language Modeling and Information Retrieval*, pages 37–41, 2001.

[56] J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. In *Proceedings of SIGIR'98*, pages 275–281. ACM, New York, 1998.

[57] A. Ratnaparkhi. A Maximum Entropy Model for Part-of-Speech Tagging. In *Proceedings of the Empirical Methods in Natural Language Conference*, 1996.

[58] K. Robertson, S.and Sparck Jones. Relevance Weighting of Search Terms. *J. Am. Soc. Inf. Sci.*, 27:129–146, 1976.

[59] S. E. Robertson. The Probability Ranking Principle in IR. *Journal of Documentation*, 33(4):294–304, 1977.

[60] S. E. Robertson, M. E. Maron, and W. S. Cooper. Probability of Relevance: A Unification of Two Competing Models for Document Retrieval. *Inf. Technol. Res. Dev.*, 1:1–21, 1982.

[61] S. E. Robertson, Walker. S., K. Sparck Jones, M. M. Hancock-Beaulieu, and M. Gatford. OKAPI at TREC-3. In *Proceedings of TREC-3*, Gaithersburg, CA, 1995. NIST.

[62] R. Rosenfeld. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. PhD thesis, Carnegie Mellon University, April 1994.

[63] R Rosenfeld. A Whole Sentence Maximum Entropy Language Model. In *in Proceedings of IEEE workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, California, December 1997.

[64] R. Rosenfeld, S. Chen, and X. Zhu. Whole-Sentence Exponential Language Models: a Vehicle for Linguistic-Statistical Integration. *Computers Speech and Language*, 15(1), 2001.

[65] G. Salton. *Automatic Text Processing*. Addison Wesley, 1989.

[66] H. Schmid. Probabilistic Part-of-Speech Tagging using Decision Trees. In *International Conference on New Methods in Language Processing*, 1994.

[67] C. Sherman. Google fires new salvo in search engine size wars.

[68] A. F. Smeaton and C. J. van Rijsbergen. Experiments on Incorporating Syntactic Processing of User Queries into a Document Retrieval Strategy. In *Proceedings of SIGIR'88*, 1988.

[69] F. Song and W. B. Croft. A General Language Model for Information Retrieval. In *Proceedings of SIGIR'99*, pages 279–280, 1999.

[70] K. Sparck-Jones. What is the Role of NLP in Text Retrieval. In T. Strzalkowski, editor, *Natural Language Information Retrieval*, pages 1–25. Kluwer, 1999.

[71] R. Srihari and W. Li. A Question Answering System Supported by Information Extraction. In *Proceedings of ANLP'00*, pages 166–172, 2000.

[72] R. Srihari, C. Niu, and W. Li. A Hybrid Approach to Named Entity and Sub-type Tagging. In *Proceedings of ANLP'00*, pages 247–254, 2000.

[73] R. K. Srihari, M. Srikanth, C. Niu, and W. Li. Use of Maximum Entropy in Back-Off Modeling for a Named Entity Tagger. In *Proceedings of the HKK Conference*, pages 159–164, June 1999.

[74] M. Srikanth, Ruiz. M. E., and R. Srihari. UB at TREC 12: HARD and Genomics Tracks. In *Proceedings of TREC-12*, Gaithersburg, CA, 2003. NIST.

[75] M. Srikanth and R. Srihari. Biterm Language Models for Document Retrieval. In *Proceedings of SIGIR'02*, pages 425–426, 2002.

[76] M. Srikanth and R. Srihari. Exploiting Syntactic Structure of Queries in a Language Modeling Approach to IR. In *in Proceedings of CIKM'03*, 2003.

[77] M. Srikanth and R. Srihari. Incorporating Query Term Dependencies in Language Models for Document Retrieval. In *Proceedings of SIGIR'03*, 2003.

[78] T. Strzalkowski, J. Perez-Carballo, J. Karlgren, A. Hulth, P. Tapanainen, and T. Lahtinen. Natural Language Information Retrieval: TREC-8 report. In *Proceedings of TREC-8*, Gaithersburg, CA, 1999. NIST.

[79] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[80] J. Wu and S. Khudanpur. Building A Topic-Dependent Maximum Entropy Language Model for Very Large Corpora. In *Proceedings of ICASSP*, 2002.

[81] R. S. Wurman, D. Sume, and L. Leifer. *Information Anxiety 2*. Que, 2000.

[82] J. Xu and W. B. Croft. Improving the Effectiveness of Information Retrieval using Local Context Analysis. *ACM TOIS*, 18(1):79–112, 2000.

[83] H. Zaragoza, D. Hiemstra, and M. Tipping. Bayesian Extension to the Langauge Model for Ad Hoc Information Retrieval. In *Proceedings of SIGIR'03*, pages 4–9. ACM, New York, 2003.

[84] C. Zhai. Exploiting Context to Identify Lexical Atoms - A Statistical View of Linguistic Context. In *Proceedings of the International and Interdisciplinary Conference on Modelling and Using Context(CONTEXT-97)2*, Rio de Janeiro, 1997.

[85] C. Zhai, W. W. Cohen, and J. Lafferty. Beyond Independent Relevance: Methods and Evaluation Metrics for Subtopic Retrieval. In *Proceedings of SIGIR*, 2003.

[86] C. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *Proceedings of SIGIR'01*, pages 334–342, 2001.