Foundations and Trends® in
Information Retrieval
Vol. 2, No. 3 (2008) 137–213

DOI: 10.1561/1500000008

now
the essence of knowledge

Statistical Language Models for Information Retrieval A Critical Review

ChengXiang Zhai

University of Illinois at Urbana-Champaign, 201 N. Goodwin, Urbana, IL 61801, USA, czhai@cs.uiuc.edu

Abstract

Statistical language models have recently been successfully applied to many information retrieval problems. A great deal of recent work has shown that statistical language models not only lead to superior empirical performance, but also facilitate parameter tuning and open up possibilities for modeling nontraditional retrieval problems. In general, statistical language models provide a principled way of modeling various kinds of retrieval problems. The purpose of this survey is to systematically and critically review the existing work in applying statistical language models to information retrieval, summarize their contributions, and point out outstanding challenges.

1

Introduction

The goal of an information retrieval (IR) system is to rank documents optimally given a query so that relevant documents would be ranked above nonrelevant ones. In order to achieve this goal, the system must be able to score documents so that a relevant document would ideally have a higher score than a nonrelevant one.

Clearly the retrieval accuracy of an IR system is directly determined by the quality of the scoring function adopted. Thus, not surprisingly, seeking an optimal scoring function (retrieval function) has always been a major research challenge in information retrieval. A retrieval function is based on a retrieval model, which formalizes the notion of relevance and enables us to derive a retrieval function that can be computed to score and rank documents.

Over the decades, many different types of retrieval models have been proposed and tested. A great diversity of approaches and methodology has developed, but no single unified retrieval model has proven to be most effective. Indeed, finding the single optimal retrieval model has been and remains a long-standing challenge in information retrieval research.

The field has progressed in two different ways. On the one hand, theoretical models have been proposed often to model relevance through inferences; representative models include the logic models [27, 111, 115] and the inference network model [109]. However, these models, while theoretically interesting, have not been able to *directly* lead to empirically effective models, even though heuristic instantiations of them can be effective. On the other hand, there have been many empirical studies of models, including many variants of the vector space model [89, 90, 91, 96] and probabilistic models [26, 51, 80, 83, 110, 109]. The vector-space model with heuristic TF-IDF weighting and document length normalization has traditionally been one of the most effective retrieval models, and it remains quite competitive as a state of the art retrieval model. The popular BM25 (Okapi) retrieval function is very similar to a TF-IDF vector space retrieval function, but it is motivated and derived from the 2-Poisson probabilistic retrieval model [84, 86] with heuristic approximations. BM25 is one of the most robust and effective retrieval functions. Another effective retrieval model is divergence from randomness which is based on probabilistic justifications for several term weighting components [1].

While both vector space models and BM25 rely on heuristic design of retrieval functions, an interesting class of probabilistic models called language modeling approaches to retrieval have led to effective retrieval functions without much heuristic design. In particular, the query likelihood retrieval function [80] with Dirichlet prior smoothing [124] has comparable performance to the most effective TF-IDF weighting retrieval functions including BM25 [24]. Due to their good empirical performance and great potential of leveraging statistical estimation methods, the language modeling approaches have been attracting much attention since Ponte and Croft's pioneering paper published in ACM SIGIR 1998 [80]. Many variations of the basic language modeling approach have since been proposed and studied, and language models have now been applied to multiple retrieval tasks such as cross-lingual retrieval [54], distributed IR [95], expert finding [25], passage retrieval [59], web search [47, 76], genomics retrieval [129], topic tracking [41, 53, 99], and subtopic retrieval [122].

This survey is to systematically review this development of the language modeling approaches. We will survey a wide range of retrieval models based on language modeling and attempt to make connections between this new family of models and traditional retrieval models. We will summarize the progress we have made so far in these models and point out remaining challenges to be solved in order to further increase their impact.

The survey is written for readers who have already had some basic knowledge about information retrieval. Readers with no prior knowledge about information retrieval will find it more comfortable to read an IR textbook (e.g., [29, 63]) first before reading this survey. The readers are also assumed to have already had some basic knowledge about probability and statistics such as maximum likelihood estimator, but a reader should still be able to follow the high-level discussion in the survey even without such background.

The rest of the survey is organized as follows. In Section 2, we review the very first generation of language models which are computationally as efficient as any other existing retrieval model. The success of these early models has stimulated many follow-up studies and extensions of language models for retrieval. In Section 3, we review work that aims at understanding why these language models are effective and why they can be justified based on relevance. In Section 4, we review work on extending and improving the basic language modeling approach. Feedback is an important component in an IR system, but it turns out that there is some difficulty in supporting feedback with the first generation basic language modeling approach. In Section 5, we review several lines of work on developing and extending language models to support feedback (particularly pseudo feedback). They are among the most effective language models for retrieval. In Section 6, we further review a wide range of applications of language models to different special retrieval tasks where a standard language model is often extended or adapted to better fit a specific application. Finally, in Section 7, we briefly review some work on developing general theoretical frameworks to facilitate systematic applications of language models to IR. We summary the survey and discuss future research directions in Section 8.

2

The Basic Language Modeling Approach

In this section, we review the basic language modeling approach (often called the query likelihood scoring method) which represents the very first generation of language models applied to information retrieval. Extensions of these models are reviewed in the next few sections.

2.1 Ponte and Croft's Pioneering Work

The language modeling approach was first introduced by Ponte and Croft in their SIGIR 98 paper [80]. In this work, they proposed a new way to score a document, later often called the *query likelihood* scoring method. The basic idea behind the new approach is simple: first estimate a language model for each document, and then rank documents by the likelihood of the query according to the estimated language model of each document. This new method was shown to be quite effective. They called this approach "language modeling approach" due to the use of language models in scoring.

The term language model refers to a probabilistic model of text (i.e., it defines a probability distribution over sequences of words). Before it was applied to retrieval, it had already been used successfully in related areas such as speech recognition [39] and machine translation [11]. In these applications, language models are used to assess what kind of word sequences are more typical according to language usages, and inject the right bias accordingly into a speech recognition system or machine translation system to prefer an output sequence of words with high probability according to the language model.

In the basic language modeling approach proposed by Ponte and Croft, the query is assumed to be a sample of words drawn according to a language model estimated based on a document (i.e., a *document language model*). We will then ask the question: which document language model gives our query the highest probability? Documents can thus be ranked based on the likelihood of generating the query using the corresponding document model. Intuitively, if a document language model gives the query a high probability, the query words must have high probabilities according to the document language model, which further means that the query words occur frequently in the document.

Formally, the general idea of the query likelihood retrieval function can be described as follows. Let Q be a query and D a document. Let θ_D be a language model estimated based on document D. We define the score of document D with respect to query Q as the conditional probability $p(Q|\theta_D)$. That is,

$$\mathrm{score}(Q,D) = p(Q|\theta_D). \tag{2.1}$$

Clearly in order to use such a model to score documents, we must solve two problems: (1) how to define θ_D? (2) how to estimate θ_D based on document D? Note that the definition of θ_D is quite critical as it would completely determine how we model the query, thus essentially determining the query representation to be adopted.

In Ponte and Croft's paper, the model θ_D has not been explicitly defined, but the final retrieval function derived suggests that the model is a multiple Bernoulli model. Formally, let $V = \{w_1,\dots,w_{|V|}\}$ be the vocabulary of the language of our documents. We can define a binary random variable $X_i \in \{0,1\}$ for each word w_i to indicate whether word w_i is present ($X_i = 1$) or absent ($X_i = 0$) in the query. Thus model θ_D would have precisely $|V|$ parameters, i.e., $\theta_D = \{p(X_i = 1|D)\}_{i\in[1,|V|]}$, which can model presence and absence of all the words in the query.

According to this model, the query likelihood can be written as:

$$p(Q|\theta_D) = \prod_{w_i \in Q} p(X_i = 1|D) \prod_{w_i \notin Q} (1 - p(X_i = 1|D)),$$

where the first product is for words in the query and the second words not occurring in the query. Computationally, the retrieval problem now boils down to estimating all the $|V|$ parameters (i.e., $p(X_i = 1|D)$) based on D; different ways to estimate the parameters would lead to different retrieval functions.

In order to estimate the multiple Bernoulli model θ_D, we would assume that document D is a sample of θ_D. If we are to interpret D as a single bit vector representing the presence and absence of each word, we would not be able to capture term frequency (TF) since a Bernoulli model only models the presence and absence of a word rather than how many times a word occurs. To capture TF, we can treat each word w_i in D as a sample from our model where only w_i has shown up and all other words are absent. Thus according to the maximum likelihood (ML) estimator, $p(X_i = 1|D)$ is equal to the relative frequency of word w_i in D, i.e.,

$$p(X_i = 1|D) = \frac{c(w_i, D)}{|D|},$$

where $c(w_i, D)$ is the count of word w_i in D and $|D|$ is the length of D (i.e., the total word counts). This derivation is discussed in detail in [69].

One problem with this ML estimator is that an unseen word in document D would get a zero probability, making all queries containing an unseen word have zero probability $p(Q|\theta_D)$. This is clearly undesirable. More importantly, since a document is a very small sample for our model, the ML estimate is generally not accurate. So an important problem we have to solve is to *smooth* the ML estimator so that we do not assign zero probability to unseen words and can improve the accuracy of the estimated language model in general.

In Ponte and Croft's model, they set the probability of an unseen word to that of the word in the whole collection of documents. Intuitively, this is to say that if we do not observe a word in the document, we would assume that the probability of such a word is the same as the probability of seeing the word in any document in the whole collection. This ensures that none of the words in the collection would get a zero probability. To further improve the robustness of smoothing, they also heuristically take the geometric mean of the ML estimate and the average term frequency in all other documents in the collection [80].

Ponte and Croft's work makes two important contributions in studying retrieval models: First, it introduces a new effective probabilistic ranking function based on query likelihood with smoothed estimate of model parameters. While the previous probabilistic models (e.g., [20, 83]) have failed to directly lead to an empirically effective retrieval function due to the difficulty in estimating parameters,[1] this new query likelihood retrieval model makes the parameter estimation problem easier to solve (see Section 3 for more discussion about this). Second, it connects the difficult problem of text representation and term weighting in IR with the language modeling techniques that have been well-studied in other application areas such as statistical machine translation and speech recognition, making it possible to exploit various kinds of language modeling techniques to address the representation problem. Such a connection was actually recognized in some early work, but no previous work has looked into the problem of how to estimate such a model accurately. For example, Wong and Yao [114] proposed to use a multinomial model to represent a document, but they just used the ML estimator and did not further study the estimation problem.

2.2 BBN and Twenty-One in TREC-7

At about the same time and apparently independent of Ponte and Croft's work, two TREC-7 participating groups, BBN [70] and Twenty-One [34], have used the same idea of scoring documents with query-likelihood but with a slightly different definition of the actual model. Instead of using multiple Bernoulli, these two groups used a multinomial distribution model, which is more commonly called a unigram language model. Such a model directly models the counts of terms and

[1] The relevance model [55] to be discussed later can be regarded as a technique to solve the parameter estimation problem in these classic probabilistic models.

is more common in other applications such as speech recognition than the multiple Bernoulli; the latter was more popular for retrieval and was already used in an earlier probabilistic retrieval model [83]. Both groups have achieved very good empirical performance in the TREC-7 evaluation using their new models.

Specifically, these two groups define θ_D as a unigram language model or multinomial word distribution, i.e., $\theta_D = \{p(w_i|D)\}_{i \in [1,|V|]}$, where $p(w_i|D)$ is the probability of word w_i. Note that as in the previous section, we use θ_D to denote a probability distribution and $p(w|D)$ to denote the probability of a word according to the distribution θ_D. However, unlike in multiple Bernoulli, where our constraint is $p(X_i = 1|D) + p(X_i = 0|D) = 1$, here our constraint is $\sum_{i=1}^{|V|} p(w_i|D) = 1$. According to such a model, the likelihood of a query $Q = q_1 ... q_m$, where q_i is a query word, would be

$$p(Q|\theta_D) = \prod_{i=1}^{m} p(q_i|D).$$

For example, a document language model θ_D might assign a probability of 0.1 to the word "computer" and 0.05 to the word "virus" (i.e., $p(computer|D) = 0.1$, $p(virus|D) = 0.05$). If our query Q is "computer virus," we would have $p(Q|\theta_D) = 0.1 * 0.05 = 0.005$. Thus intuitively, the more frequently a query word occurs in document D, the higher the query likelihood would be for D, capturing the basic TF retrieval heuristic [24].

As in the case of multiple Bernoulli, the retrieval problem is now reduced to the problem of estimating the language model θ_D (i.e., $p(w|D)$ for each word w). Once again, the issue of smoothing the ML estimate is critical. In both groups' work, θ_D is smoothed by interpolating the ML estimate with a background language model estimated using the entire collection:

$$p(w|D) = (1 - \lambda)\frac{c(w,D)}{|D|} + \lambda p(w|C),$$

where $p(w|C)$ is a collection (background) language model estimated based on word counts in the entire collection and $\lambda \in [0,1]$ is a smoothing parameter.

The two groups used a slightly different formula for estimating $p(w|C)$; BBN used the normalized total counts of a word in the collection while Twenty-One used the normalized total number of documents containing a word [34, 70]. The general idea is similar, though, and it is also similar to what Ponte and Croft used in their estimate of θ_D.

These two groups also went beyond the basic query likelihood scoring formula to introduce a document prior $p(D)$ using Bayes formula, thus suggesting that we essentially score a document based on the conditional probability $p(D|Q)$:

$$p(D|Q) = \frac{p(Q|D)p(D)}{p(Q)} \propto p(Q|D)p(D). \tag{2.2}$$

Note that in this derivation, we have not specified how to interpret $p(Q|D)$. One way is to interpret it as $p(Q|\theta_D)$, which would give us precisely the basic query likelihood scoring formula originally introduced in [80]. However, other interpretations may also be possible (e.g., the translation model [4]).

The document prior $p(D)$ (which should be distinguished from $p(\theta_D)$) can be useful for introducing additional retrieval criteria to favor documents with certain features, and indeed has been explored in [47, 49, 58]. This prior presumably can also be added to the query likelihood formula proposed by Ponte and Croft. Thus this formulation is a more general formulation of the basic language modeling approach than the query likelihood retrieval function proposed by Ponte and Croft. A similar formulation was also given in [75], where Ng also discussed other issues including how to estimate the smoothing parameter with pseudo feedback.

There are two additional points worth mentioning: First, the BBN group presented their model as a Hidden Markov Model (HMM) [82]; the HMM makes smoothing more explicit and also has the potential to accommodate more sophisticated models, particularly combining different representations [70]. Second, the Twenty-One group revealed the connection of their language model with traditional retrieval heuristics such as TF-IDF weighting [96] and document length normalization [34, 30], which offers an intuitive justification for this new retrieval model. A more general derivation of this connection can be found in [124].

2.3 Variants of the Basic Language Modeling Approach

The basic language modeling approach (i.e., the query likelihood scoring method) can be instantiated in different ways by varying (1) θ_D (e.g., multiple Bernoulli or multinomial), (2) estimation methods of θ_D (e.g., different smoothing methods), or (3) the document prior $p(D)$. Indeed, this has led to many variants of this basic model, which we now briefly review.

Although the original work by Ponte and Croft used the multiple Bernoulli model, it has not been as popular as the multinomial model. One reason may be because the latter can capture the term frequency in documents (as well as the query) more naturally than the former; indeed, the multiple Bernoulli model clearly ignores query term frequencies and is also somewhat unnatural to incorporate TF in the documents. Note that both models make some independence assumption about term occurrences, but their assumptions are different. In multiple Bernoulli model, the presence/absence of a term is assumed to be independent of that of other terms, whereas in multinomial model, every word occurrence is assumed to be independent, including the multiple occurrences of the *same* term. Since once an author starts to use a term in a document, the author tends to use the term again, treating multiple occurrences of the same term as independent can cause "over counting" of the occurrences of a term. Recently some new models (e.g., Dirichlet compound multinomial [62]) have been proposed to address this problem and model word burtiness; they may potentially lead to better retrieval models. The multiple Bernoulli model and multinomial model were compared empirically earlier in the context of text categorization [64] and later in the context of query likelihood for retrieval [52, 97]. The current conclusions seem to be that multinomial model is better, but more evaluation is needed to make more definitive conclusions.

Recently, a Poisson model was proposed as an alternative for the query likelihood retrieval function [65] and some promising results have been achieved. One potential advantage of multiple Bernoulli over multinomial is the possibility of naturally smoothing the model for each term independently (because each term is treated as an independent event),[2] which provides flexibility for optimizing smoothing at a per-term basis, while multinomial can naturally capture term frequencies in the query, which are ignored in multiple Bernoulli. Poisson model can accommodate both flexible smoothing and modeling term frequencies, making it a very interesting model to further study.

Another variation is to relax the independence assumptions made in the basic model to capture some limited dependency such as through bigram language models. We will review this line of work and other extensions in Section 4.

Estimation of θ_D is quite critical for this family of models, and a particularly important issue is how to smooth the maximum likelihood estimate which assigns zero probability to unseen words. Many different smoothing methods have been used. In addition to those mentioned earlier in our discussion, there are many other smoothing methods that can be applied (see, e.g., [17, 45]). Zhai and Lafferty [124] empirically compared three different smoothing methods, including Jelinek-Mercer (fixed coefficient interpolation) [40], Dirichlet prior [61], and absolute discounting [74], on several standard test collections. Most of these smoothing methods end up interpolating the original maximum likelihood estimate with the collection background language model in some way. Despite this similarity, different smoothing methods can perform differently. It was found that in general, Dirichlet prior smoothing works the best, especially for keyword queries (nonverbose queries). The reason may be because it adjusts the amount of smoothing according to the length of a document in a reasonable way (longer documents get less smoothing). Furthermore, interpolation-based smoothing methods all work better than backoff smoothing methods [17, 44], even though the latter works well for speech recognition, which is likely due to the lack of an IDF effect in backoff smoothing. This point will be further elaborated in Section 3.2.

[2] One can also let the Dirichlet prior smoothing parameter μ take a term-specific value μ_i for term w_i to achieve term-specific smoothing, for multinomial, but this is not as natural as in the case of multiple Bernoulli.

The Dirichlet prior smoothing method can be derived by using Bayesian estimation (instead of ML estimation) with a Dirichlet conjugate prior [61, 125], and the formula is as follows:

$$p(w|D) = \frac{c(w,D) + \mu p(w|C)}{|D| + \mu},$$

where $p(w|C)$ is a background (collection) language model and μ is a smoothing parameter, which can be interpreted as the total number of pseudo counts of words introduced through the prior. The Dirichlet prior smoothing method has now become a very popular smoothing method for smoothing language models in an IR task.

The study by Zhai and Lafferty has also shown that retrieval performance can be quite sensitive to the setting of smoothing parameters and suggested that smoothing plays two different roles in the query likelihood retrieval formula, an issue we will further discuss later.

All these smoothing methods discussed so far are simple in the sense that different documents are smoothed using the same collection language model. Intuitively, each document can have its own "reference language model" for smoothing, and there has been work done in this direction. We will review this line of work in Section 4.

The document prior $p(D)$ can naturally incorporate any static ranking preferences of documents (i.e., ranking preferences independent of a query) such as PageRank scores or other document features. In this line of the work, Kraaij and coauthors [47] successfully leveraged this prior to implement an interesting Web search heuristic for named page finding. Their idea is to prefer pages with shorter URLs since an entry page tends to have a shorter URL. They used some training data to estimate the prior $p(D)$ based on URL lengths, and showed that this prior can improve search performance significantly [47]. Li and Croft [58] studied how to leverage the document prior $p(D)$ to implement time-related preferences in retrieval so that a document with a more recent time would be preferred. This strategy has been shown to be effective for a particular set of "recency queries." In a study by Kurland and Lee [49], a PageRank score computed using induced links between documents based on document similarity has been used as a prior to improve retrieval accuracy. In [132] priors to capture document quality are shown to be effective for improving the accuracy of the top-ranked documents in ad hoc web search.

2.4 Summary

In this section, we reviewed the basic language modeling approach, which is roughly characterized by the use of query likelihood for scoring and simple smoothing methods based on a background collection language model. Such a basic approach (especially with Dirichlet prior smoothing) has been shown to be as effective as well-tuned existing retrieval models such as pivoted length normalization and BM25 [24]. Retrieval functions in this basic language modeling approach can generally be computed as efficiently as any standard TF-IDF retrieval model with the aid of an inverted index.[3]

[3] This point will be further elaborated in Section 3.2.

3

Understanding Query Likelihood Scoring

Although the query likelihood retrieval method has performed well empirically, there were questions raised regarding its foundation as a retrieval model, particularly its connection with the key notion in retrieval — relevance [98]. Indeed, none of the early work has provided a rigorous treatment of the language model θ_D, nor has it provided a solid connection between query likelihood and relevance. Thus it is unclear how we should interpret θ_D: is it a model for documents or for queries? One may also question whether such models have just happened to perform well, but without any solid relevance-based foundation.

3.1 Relevance-based Justification for Query Likelihood

The superficial justification based on Equation (2.2) suggests the following relevance-based interpretation: Suppose that there is precisely one relevant document, and the observed query has been "generated" using that relevant document. We can then use the Bayes' Rule to infer which document is "that relevant document" based on the observed query. This leads to Equation (2.2), which boils down to scoring with $P(Q|D)$ under the assumption of a uniform prior $p(D)$. Unfortunately, such a “single relevant document” formulation raises many questions as discussed in [98].

To better understand the retrieval foundation of the query likelihood method, Lafferty and Zhai [51] offered a more general relevance-based derivation of the query likelihood method. Specifically, they show that the query likelihood retrieval function can be justified in a similar way as the classical probabilistic retrieval model based on the probability ranking principle [85].

The starting point of the derivation is the conditional probability $p(R = 1|Q,D)$ ($R \in \{0,1\}$ is a binary relevance random variable) which is the probability that document D is relevant to query Q. The probability ranking principle provides a justification for ranking documents for a query based on this conditional probability. This is equivalent to ranking documents based on the odds ratio, which can be further transformed using Bayes’ Rule:

$$O(R = 1|Q,D) = \frac{p(R = 1|Q,D)}{p(R = 0|Q,D)} \propto \frac{p(Q,D|R = 1)}{p(Q,D|R = 0)}. \tag{3.1}$$

At this point, we may decompose the joint probability $P(Q,D|R)$ in two different ways:

(1) document generation: $p(Q,D|R) = p(D|Q,R)p(Q|R)$, and
(2) query generation: $p(Q,D|R) = p(Q|D,R)p(D|R)$.

With document generation, we have

$$O(R = 1|Q,D) \propto \frac{p(D|Q,R = 1)}{p(D|Q,R = 0)}.$$

Thus the ranking is equivalent to the ranking given by the classical Robertson–Sparck-Jones probabilistic model [83] if we define $P(D|Q,R)$ as multiple Bernoulli models [51]: See [88] for an in-depth discussion of other variants such as multinomial and Poisson as well as the relationship between different variants.

With query generation, we have

$$O(R = 1|Q,D) \propto \frac{p(Q|D,R = 1)}{p(Q|D,R = 0)} \frac{p(R = 1|D)}{p(R = 0|D)}.$$

If we make the assumption that $p(Q|D,R=0) = p(Q|R=0)$ (i.e., the distribution of "nonrelevant queries" does not depend on the particular document, which is not a very strong assumption), we obtain

$$O(R=1|Q,D) \propto p(Q|D,R=1)\frac{p(R=1|D)}{p(R=0|D)}.$$

The term $\frac{p(R=1|D)}{p(R=0|D)}$ can be interpreted as a prior of relevance on a document, which can be estimated based on additional information such as the links pointing to D in a hypertext collection. Without such extra knowledge, we may assume that this term is the same across all the documents, which gives the following relation [51][1]

$$O(R=1|Q,D) \propto p(Q|D,R=1),$$

which provides a relevance-based justification for the query likelihood scoring method.

Note that the query likelihood derived above, i.e., $p(Q|D,R)$, has an additional relevance variable R in the conditional probability, which is essential to obtain a relevance-based justification for query likelihood scoring and gives a clear interpretation of the θ_D mentioned earlier. Specifically, it suggests that the probability $p(Q|D)$ used in all the query likelihood scoring methods should be interpreted as $p(Q|D,R=1)$, which intuitively means *the probability that a user who likes document D would pose query Q*.

Clearly this does not mean that the user can only like one document, thus there is no concern about the "single relevant document." Furthermore, this also suggests that θ_D is really a model for what kind of queries would be posed if a user wants to retrieve document D. When we estimate θ_D using a document D as observation, we are essentially using words in D to approximate the queries a user might pose to retrieve D, which is reasonable but not ideal. For example, anchor text describing a link to document D can be a better approximation of the queries a user might pose to retrieve D, and can thus be leveraged to improve the estimate of θ_D as explored in [73].

[1] A stronger assumption $p(R,D) = p(R)p(D)$ has been used in [51] to derive this relation.

3.2 Query Likelihood, Smoothing, and TF-IDF Weighting

In another line of work attempting to understand why query likelihood scoring is effective as a retrieval method, Zhai and Lafferty studied the robustness of query likelihood scoring and examined how retrieval performance is affected by different strategies for smoothing [121, 124, 126]. Through comparing several different smoothing methods, they have observed: (1) retrieval performance is sensitive to the setting of smoothing parameters and the choice of smoothing methods; (2) the sensitive patterns are different for keyword queries (all words are content-carrying keywords) and verbose queries (queries are sentences describing the information need, thus contain many common nondiscriminative words).

The first observation suggests that while heuristic term weighting in traditional retrieval models has been replaced with language model estimation (particularly smoothing) in the query likelihood approach, we have not been able to escape from the need for heuristic tuning of parameters since nonoptimal smoothing can degrade retrieval performance significantly. However, compared with TF-IDF weighting parameters, a smoothing parameter is more meaningful from the view point of statistical estimation. Indeed, completely automatic tuning of the smoothing parameters is shown to be possible in [125] and the performance with automatic parameter setting is comparable to the optimal performance achieved through manual tuning.

The second observation suggests that smoothing plays two distinct roles in the query likelihood scoring methods: one obvious role is to address the data sparseness issue (since a document is a small sample) and improve the accuracy of the estimated language model; the other nonobvious role is to model the noisy (nondiscriminative) words in the query. It is conjectured that it is this second role that has caused the different sensitivity patterns for keyword and verbose queries; indeed since the modeling of noise in queries is much more critical for verbose queries than keyword queries, it is not surprising that additional smoothing is often needed (for the second role) to achieve optimal performance for verbose queries than keyword queries as observed in [126].

This second role of smoothing is also closely related to a general connection between smoothing with a background language model and the IDF effect in the query likelihood scoring formula. In [124], it is shown that if we smooth a document language model with a general smoothing scheme where an unseen word w in document D would have a probability proportional to the probability of the word given by a collection language model (i.e. $p(w|D) = \alpha_D p(w|C)$ with a parameter α_D to control the amount of smoothing), the query likelihood scoring function can be rewritten as follows:

$$\log p(Q|D) = \left[\sum_{i:c(q_i,D)>0} \log \frac{p_s(q_i|D)}{\alpha_D\, p(q_i|C)} \right] + m \log \alpha_D + \sum_{i=1}^{m} \log p(q_i|C),$$

where $p_s(q_i|D)$ is the smoothed probability of a *seen* query word q_i and m is the query length.

Since the last term does not affect ranking, it can be ignored for ranking. As a result, we see that the formula essentially involves a sum of term weights over all the matched query terms in the document, just as in any other traditional retrieval function. Moreover, each matched term contributes a TF-IDF like weight. In this sense, the query likelihood retrieval function simply offers an alternative way of implementing TF-IDF weighting and document length normalization heuristics. In particular, the IDF effect is achieved through having $p(q_i|C)$ in the denominator of the weighting function. This means that through smoothing, we *implicitly* penalize words that are common in the collection (with high $p(q_i|C)$). This also explains why we can model the noise in the query through more aggressive smoothing. See [125] for more discussion about this.

The equation above also shows that computing the query likelihood scoring function using *any* smoothing method based on a collection language model is as efficient as computing a traditional retrieval function such as the pivoted length normalization function of the vector space model [96] with an inverted index. Indeed, the query likelihood retrieval function with several different smoothing methods has been implemented in this way in the Lemur toolkit (http://www.lemurproject.org/), which is the main retrieval toolkit currently available for experimenting with language modeling retrieval approaches.

These understandings provide an empirical explanation for why the query likelihood retrieval function is reasonable from retrieval perspective in the sense that it can be regarded as just another way of implementing TF-IDF weighting and document length normalization. They also suggest that the implementation of IDF heuristic in this approach is not as direct as in a traditional model, leading some researchers to have explored alternative ways to incorporate IDF [35].

4

Improving the Basic Language Modeling Approach

In Section 2, we restricted the discussion to the family of models that use simple smoothing methods based on a background language model; their efficiency is comparable to any traditional TF-IDF model. In this section, we review some improvements to the basic language modeling approach, which often outperform the basic approach, but also tend to demand significantly more computation than the basic approach. All these improvements remain in the family of query-likelihood scoring, which distinguishes them from the other models to be reviewed in the next section.

4.1 Beyond Unigram Models

A natural extension of the basic query likelihood method is to go beyond unigram language models. Unlike unigram language models where the occurrences of words are assumed to be completely independent (an assumption obviously not holding), these models can capture some dependency between words.

For example, Song and Croft [97] have studied using bigram and trigram language models. In a bigram language model, the generation of a current word would be dependent on the previous word generated, thus it can potentially capture the dependency of adjacent words (e.g., phrases). Specifically, the query likelihood would be

$$p(Q|D) = p(q_1|D)\prod_{i=2}^{m} p(q_i|q_{i-1}, D),$$

where $p(q_i|q_{i-1}, D)$ is the conditional probability of generating q_i after we have just generated q_{i-1}.

Such n-gram models capture dependency based on word positions. Other work has attempted to capture dependency based on grammar structures [28, 72, 100, 102, 101]. In all these approaches, the retrieval formula eventually boils down to some combination of scores from matching units larger than single words (e.g., bigrams, head-modifier pairs, or collocation pairs). While these approaches have mostly shown benefit of capturing dependencies, the improvement tends to be insignificant or at least not so significant as some other extensions that can achieve some kind of pseudo feedback effect. (These other extensions will be reviewed in the next section.) One reason for these nonexciting results may be because as we move to more complex models to capture dependency, our data becomes even more sparse, making it difficult to obtain accurate estimation of the model. The general observation on these models is consistent with what researchers have observed on some early effort on applying natural language processing techniques to improve indexing, notably phrase-based indexing [23, 56, 104, 120].

A more successful retrieval model that can capture limited dependencies is the Markov Random Field model proposed in [68]. This model is a general discriminative model where arbitrary features can be combined in a retrieval function. In most of the applications of such a model, the features are typically the scores of a document with respect to a query using an existing retrieval function such as the query likelihood, thus the Markov Random Field model essentially serves as a way to combine multiple scoring strategies and scoring with multiple representations. In particular, it has been shown that one can combine unigram language modeling scoring with bigram scoring as well as scoring based on word collocations within a small window of text.

Such a combination achieves better retrieval accuracy than using only unigram scoring [68].

4.2 Cluster-based Smoothing and Document Expansion

Smoothing every document with the *same* collection language model is intuitively not optimal since we essentially assume that all the unseen words in different documents would have similar probabilities. Ideally, we should use some document-dependent "augmented text data" that can more accurately reflect the content of the document under consideration. With such reasoning, several researchers have attempted to exploit the corpus structure to achieve such document-specific smoothing.

The work in this line can be grouped into two categories: (1) Cluster documents and smooth a document with the cluster containing the document. (2) For each document, obtain the most similar documents in the collection and then smooth the document with the obtained "neighbor documents."

In Liu and Croft [60], documents are clustered using a cosine similarity measure, and each document is smoothed with the cluster containing the document by interpolating the original maximum likelihood estimate $p(w|D)$ with a cluster language model $p(w|Cluster)$, which is further smoothed by interpolating itself with a collection language model $p(w|C)$. Such a model is shown to outperform the baseline smoothing method that only uses the collection language model for smoothing. However, the improvement is mostly insignificant. One possible reason may be because the two roles of smoothing have been mixed, thus if the parameters are not set appropriately, then smoothing using cluster-based language model may actually end up penalizing terms common in the cluster due to the IDF effect of smoothing, thus lowering the scores of documents matching terms in the cluster. In Kurland and Lee [48], the authors presented a general strategy for exploiting the cluster structure to achieve an effect similar to smoothing document language models with cluster language models; document language models are not explicitly smoothed with a cluster language model, but a document is scored based on a weighted combination of its regular query likelihood score with the likelihood of the query given the clusters containing the document.

A soft clustering strategy has been adopted to smooth document language models through using the Latent Dirichlet Allocation (LDA) model to do clustering [113]. With this model, we allow a document to be in multiple topics (roughly like document clusters, but characterized by unigram language models) with some uncertainties. Thus smoothing of a document can involve an interpolation of potentially many clusters; this is different from [60], where just one cluster is used for smoothing. Results reported in [113] are quite encouraging.

A problem with smoothing a document using a cluster is that the cluster is not necessarily a good representation of similar documents to the document to be smoothed. This is clearly the case when the document is at the boundary of the cluster. To address this problem, Tao and others [106] proposed to construct a document-specific "neighborhood" in the document space, essentially to form a cluster for each document with the document at the center of the cluster. Intuitively, such a neighborhood contains the documents that are most similar to the document, thus serves well for smoothing. To further improve the robustness of the smoothing method, the authors assign weights to the neighbors based on a cosine similarity measure so that a document farther away would contribute less to smoothing. They then use the probabilistic neighborhood to smooth the *count* of a word by interpolating the original count in the document with a weighted sum of counts of the word in the neighbor documents to obtain a smoothed count for each word. Such smoothed counts thus represent an "expanded document," and are then used as if they were the true counts of the words in the document for further smoothing with a collection language model. Experiment results show that such a document expansion method not only outperforms the baseline simple smoothing method (i.e., with only a collection language model), but also outperforms the cluster-based smoothing method proposed in [60]. Moreover, it can be combined with pseudo feedback to further improve performance [106].

In [93], this neighborhood-based document expansion method is further extended to allow for smoothing with *remotely* related documents through probabilistic propagation of term counts. This new smoothing method is shown to outperform the simple smoothing methods using a collection language model. It also achieves consistently better precision in the top-ranked documents than both cluster-based and document expansion smoothing methods. But interestingly, it has a worse mean average precision than the latter, indicating room for further research to improve this smoothing method.

A general optimization framework, which covers mostly all the work mentioned above as special cases, is proposed in [67]. In this work, a general objective function is explicitly defined for smoothing language models over graph structures, thus offering a general principled framework for smoothing. Several new smoothing methods have been derived using the framework with some outperforming the state of the methods [67].

4.3 Parsimonious Language Models

All the methods for smoothing discussed so far end up interpolating counts of words in various documents, thus the estimated document language model generally assigns high probabilities to frequent words including stop words. From retrieval perspective, we would like our model to be more discriminative (i.e., IDF heuristic). While smoothing with a collection language model can achieve the needed discrimination indirectly, one may also attempt to do it more directly. Motivated by this reasoning, a "distillation" strategy with a two-component mixture model was proposed in [121], where a query or a document is assumed to be generated from a mixture model involving two components: one is a fixed background (collection) language model and the other a topic language model to be estimated. If we estimate the topic language model by fitting such a two-component mixture model to some text sample (e.g., query or document), the common words would be easily "explained" by the background model; as a result, the estimated topic model would be more discriminative and tend to assign high probabilities to content-carrying words which do not have high probabilities according to the background model. The query distillation experiments in [121] have shown positive results from using the distillation strategy.

Such a distillation strategy was further generalized in [35] to be used in all stages of retrieval, including indexing stage, query stage, and feedback stage. In all cases, the basic idea is to use a background language model to “factor out” the nondiscriminative “background words” from a language model. The authors call such language models *parsimonious language models*. Unfortunately, such parsimonious models have not shown significant improvement in retrieval accuracy, though they can be useful for reducing the index size [35]. This result appears to be counter-intuitive. There could be two possible explanations: (1) The current language modeling retrieval approach already has some “built-in” IDF heuristic (i.e., through interpolation with a collection language model), so making the estimated model more discriminative would not add more benefit. (2) There may be complicated interactions between smoothing and term weighting, so with a more discriminative language model, we may need to adjust smoothing accordingly as well. Further study of such models is needed to understand their potential better.

4.4 Full Bayesian Query Likelihood

In all the work we have discussed so far, we estimate θ_D using a point estimator, which means we obtain our best guess of θ_D. Intuitively, there are uncertainties associated with our estimate, and our estimate may not be accurate. A potentially better method is thus to consider this uncertainty and use the posterior distribution of θ_D (i.e., $p(\theta_D|D)$) to compute the query likelihood. Such a full Bayesian treatment was proposed and studied in Zaragoza and others [119].

Their new scoring function is

$$p(Q|D) = \int p(Q|\theta_D)p(\theta_D|D)d\theta_D.$$

The regular query likelihood scoring formula can be seen as a special case of this more general query likelihood when we assume that $p(\theta_D|D)$ is entirely concentrated at one single point.

Although the integral looks intimidating, it actually has a closed form solution when we use a conjugate prior for computing the posterior distribution $p(\theta_D|D)$, making it relatively efficient to compute this likelihood. Indeed, the scoring formula is not much more expensive than a scoring formula using simple smoothing [119].

Unfortunately, empirical evaluation shows that this new model, while theoretically very interesting, does not outperform the simple query likelihood function significantly. However, when this new model is combined with linear interpolation smoothing, the performance is better than any other combinations of existing smoothing methods. This may suggest that the new model cannot model the query noise very well, thus it can be substantially improved when it is combined with the linear interpolation smoothing to obtain the extra smoothing needed for modeling query noise. As the authors pointed out, it would be interesting to further study how to model the query noise using a full Bayesian model.

4.5 Translation Model

The work mentioned so far is all essentially based on the same query likelihood scoring function which performs the retrieval task through *exact* keyword matching in a way similar to a traditional retrieval model. In order to allow *inexact* matching of semantically related words and address the issues of synonym and polysemy, Berger and Lafferty proposed a very important extension to the basic exact matching query likelihood function by allowing the query likelihood to be computed based on a *translation model* of the form $p(u|v)$, which gives the probability that word v can be "semantically translated" to word u [4].

Formally, in this new model, the query likelihood is computed in the following way:

$$p(Q|D) = \prod_{i=1}^{m} \sum_{w \in V} p(q_i|w)p(w|D),$$

where $p(q_i|w)$ is the probability of "translating" word w into q_i. This translation model can be understood by imagining a user who likes document D would formulate a query in two steps. In the first, the user would sample a word from document D; in the second, the user would "translate" the word into possibly another different but semantically related word.

It is easy to see that if $p(q_i|w)$ only allows a word to be translated into itself, we would recover the simple exact matching query likelihood. In general, of course, $p(q_i|w)$ would allow us to translate w to other semantically related words by giving those other words a nonzero probability. This enables us to score a document by counting the matches between a query word and a different but semantically related word in the document.

A major challenge here is how to obtain the translation model $p(q_i|w)$. The best training data for estimating this translation model would be many relevance judgments for all the documents. Unfortunately we generally do not have such training data available. As an approximation, Berger and Lafferty used a heuristic method to generate some synthetic query-document pairs for training the translation model. Using this method, they have shown that the translation model can improve retrieval performance significantly over the baseline exact matching query likelihood [4].

An alternative way of estimating the translation model based on document titles was proposed in [42], which has also been shown to be effective. Furthermore, WordNet and co-occurrences of words have been exploited to define the translation model $p(q_i|w)$ in [14], and improvement of performance is observed.

Another challenge in using such a model in a practical system is how to improve the scoring efficiency as we now have to consider many other words for possible matchings with each query word. Indeed, evaluation of this method in TREC-8 has revealed that there are significant challenges in handling efficiently the large number of parameters and scoring all the documents [5].

Despite these challenges, the translation model provides a principled way of achieving "semantic smoothing" and enables semantic matching of related words. It thus makes an important contribution in extending the basic query likelihood retrieval model. Such a model has later been used successfully in applying language models to cross-lingual information retrieval [118].

The cluster-based query likelihood method proposed in [48] can also be regarded as a form of a translation model where the whole document is "translated" into a query as a single unit through a set of clusters, giving the following query likelihood formula:

$$p(Q|D) = \sum_{G_i \in G} p(Q|G_i)p(G_i|D),$$

where G_i is a cluster of documents and G is a pre-constructed set of clusters of documents in the collection. This method has shown some improvement over the simple query likelihood method when combined with the simple query likelihood method, but does not perform well alone. Since the translation of a document into a cluster G_i causes loss of information, matching based on the clusters *alone* may not be discriminative enough to distinguish relevant documents from nonrelevant ones, even though such a matching can potentially increase recall due to the allowed inexact matching of terms. This may explain why such methods alone often do not perform well, but they would perform much better when they are combined with a basic model that can supply the needed word-level discrimination. Similar observations have also been made in [38] where Probabilistic Latent Semantic Indexing (PLSI) was used to learn a lower dimension representation of text in terms of probabilistic topics. PLSI will be discussed further in Section 6.8.

4.6 Summary

In this section, we reviewed a number of models that all attempted to extend the basic query likelihood retrieval method in various ways. They are often substantially more expensive to compute than the basic model. Many of the extensions have not really led to significant improvement over the basic model. Given their complexity and the relative insignificant improvement (compared with models to be reviewed in the next section), most of these models have not found widespread applications. However, some document-specific smoothing methods have been shown to improve performance significantly, and the computation can often be done offline at the indexing stage. So it is still feasible to use these methods in a large-scale application system. Also, the translation model has later been applied to cross-lingual IR tasks successfully.

5

Query Models and Feedback in Language Models

In the query likelihood retrieval model, we are mostly concerned with estimating a document language model θ_D. One limitation of this scoring method is that it is unnatural and hard to support feedback mainly because the query is assumed to be a sample of a language model, thus it is unclear how we should interpret any expanded/modified query. To address this problem, a new scoring strategy based on computing the Kullback–Leibler (KL) divergence of a document language model and a query language model has been introduced. The KL-divergence retrieval model can naturally support feedback by casting it as a problem of estimating a query language model (or relevance model) based on feedback information. In this section, we review this development.

5.1 Difficulty in Supporting Feedback with Query Likelihood

Feedback is an important technique to improve retrieval accuracy. Both relevance feedback and pseudo feedback have been well-supported in traditional models (e.g., Rocchio [87] for the vector space model and term re-weighting for the classical probabilistic model [83]). Naturally, in the early days when the query likelihood scoring method was introduced, people also explored feedback [70, 75, 79].

However, unlike in the traditional models where feedback can be naturally accommodated, in the query likelihood retrieval method, it is rather awkward to support feedback. The problem is caused by the fact that in all the query likelihood retrieval methods, the query is regarded as a sample of some kind of language model. Thus it would not make much sense to talk about improving the query by adding additional terms and/or adjusting weights of those terms as done in the vector space model [87], nor is it natural to use the feedback documents from a user or from pseudo feedback to improve our estimate of models (i.e., improving θ_D) as done in the classical probabilistic model [83].

Due to this difficulty, early work on achieving feedback using the query likelihood scoring method tends to be quite heuristic, and the techniques used are often not as elegant as the query likelihood method itself. For example, in [79], terms with high probabilities in the feedback documents but low probabilities in the collection are selected using a ratio approach as additional query terms. While this method generally performs well (similarly to Rocchio [87]), the ratio approach is conceptually restricted to the view of query as a set of terms, so it cannot be applied to the more general case when the query is considered as a sequence of terms in order to incorporate the frequency information of a query term. Also, the influence of feedback cannot be controlled through term weighting; a term is either added to the query or not. Similar strategies for heuristic expansion of queries were also studied in Miller and others [70] and [75]. But in all these approaches, it is no longer conceptually clear how to interpret the expanded query in a probabilistic way.

Several studies [31, 32, 75, 125] have used feedback documents to optimize the smoothing parameter or query term re-weighting. While these methods do not cause conceptual inconsistency, they also do not achieve full benefit of feedback due to the limited use of feedback information.

5.2 Kullback–Leibler Divergence Retrieval Model

The difficulty in supporting feedback with query likelihood scoring has motivated the development of a more general family of probabilistic similarity models called Kullback–Leibler (KL) divergence retrieval model [50, 123]. In this model, we define two different language models, one for a query (θ_Q) and one for a document (θ_D). That is, we will assume that the query is a sample observed from a query language model θ_Q (which presumably represents a user's information need), while the document is a sample from a document language model θ_D (which represents the topic/content of a document). We can then use the KL-divergence of these two models to measure how close they are to each other and use their distance (indeed, negative distance) as a score to rank documents. This way, the closer the document model is to the query model, the higher the document would be ranked.

Intuitively, such a scoring strategy is very similar to the vector-space model except that we now have probabilistic representation of text and work with a probabilistic distance/similarity function. Formally, the score of a document D with respect to a query Q is given by:

$$\begin{aligned} s(D,Q) &= -D(\theta_Q||\theta_D) && (5.1)\\ &= -\sum_{w \in V} p(w|\theta_Q)\log\frac{p(w|\theta_Q)}{p(w|\theta_D)} && (5.2)\\ &= \sum_{w \in V} p(w|\theta_Q)\log p(w|\theta_D) - \sum_{w \in V} p(w|\theta_Q)\log p(w|\theta_Q) && (5.3)\end{aligned}$$

Since the last term is query entropy and does not affect ranking of documents, ranking based on negative KL-divergence is the same as ranking based on negative cross entropy $\sum_{w \in V} p(w|\theta_Q)\log p(w|\theta_D)$.[1]

With this model, the retrieval task is reduced to two subtasks — estimating θ_Q and θ_D, respectively. The estimation of document model θ_D is similar to that in the query likelihood retrieval model, but the estimation of query model θ_Q offers interesting opportunities of leveraging feedback information to improve retrieval accuracy. Specifically,

[1] Note that although the two ways to rank documents are equivalent in the *ad hoc* retrieval setting, where we always compare documents for the *same* query and can thus ignore any document-independent constant, the KL-divergence and cross entropy would generate quite different results when we compare scores across *different* queries as in the case of filtering or topic detection and tracking [46]. Specifically, one measure may generate scores more comparable across queries than the other, depending on whether including the query entropy makes sense. For a detailed analysis of this difference and alternative ways of normalizing scores, see [46].

feedback information can be exploited to improve our estimate of θ_Q. Such a feedback method is called *model-based feedback* in [123].

On the surface, the KL-divergence model appears to be quite different from the query likelihood method. However, it turns out that it is easy to show that the KL-divergence model covers the query likelihood method as a special case when we use the empirical query word distribution to estimate θ_Q [50], i.e.,

$$p(w|\theta_Q) = \frac{c(w,Q)}{|Q|}.$$

In this sense, the KL-divergence model is a generalization of the query likelihood scoring method with the additional advantage of supporting feedback more naturally.

This KL-divergence retrieval model was first proposed in [50] within a risk minimization retrieval framework, which introduces the concept of query language model (in additional to the document language model) and models the retrieval problem as a statistical decision problem [50, 121, 127]. However, KL-divergence had previously been used for distributed information retrieval [117].

By truncating the query model θ_Q to keep only high probability words and renormalizing it, we can score a KL-divergence model efficiently. Indeed, we may rewrite the scoring function in the same way as in the case of query likelihood to obtain a scoring formula essentially involving a sum over the terms with nonzero probabilities for both θ_Q and θ_D [76]:

$$score(D,Q) \overset{\text{rank}}{=} \sum_{w \in D} p(w|\theta_Q) \log \frac{p_s(w|\theta_D)}{\alpha_D p(w|C)} + \log \alpha_D. \tag{5.4}$$

Thus the generalization of query likelihood as KL-divergence does not really incur much extra computational overhead; it is generally regarded as a state-of-the-art retrieval model based on language modeling.

5.3 Estimation of Query Models

With the KL-divergence retrieval model, feedback can be achieved through re-estimating the query model θ_Q based on feedback information.

Several methods have been proposed to perform such model-based feedback in the pseudo feedback setting. Interestingly, the relevance feedback setting appears to have not attracted that much attention, though these methods can presumably also be applied to relevance feedback. However, sometimes these methods may need to be adapted appropriately to handle relevance feedback; in particular, feedback based on only negative information (i.e., nonrelevant information) remains challenging with the KL-divergence retrieval model and additional heuristics may need to be used [112]. This is in contrast to the document-generation probabilistic models such as the Robertson–Sparck Jones model [83] which can naturally use negative examples to improve the estimate of the nonrelevant document model.

5.3.1 Model-based Feedback

Zhai and Lafferty [123] proposed two methods for estimating an improved query model θ_Q using feedback documents. Both methods follow the basic idea of interpolating an existing query model (e.g., one estimated based on the empirical query word distribution) with an estimated feedback topic model.

Specifically, let θ_Q be the current query model and θ_F be a feedback topic model estimated based on (positive) feedback documents $F = \{d_1,\ldots,d_n\}$. The updated new query model θ'_Q is given by

$$p(w|\theta'_Q) = (1-\alpha)p(w|\theta_Q) + \alpha p(w|\theta_F),$$

where $\alpha \in [0,1]$ is a parameter to control the amount of feedback. The two methods differ in the way of estimating θ_F with F.

One approach uses a two-component mixture model to fit the feedback documents where one component is a fixed background language model $p(w|C)$ estimated using the collection and the other is an unknown, to-be-discovered topic model $p(w|\theta_F)$. Essentially, the words in F are assumed to fall into two kinds: (1) background words (to be explained by $p(w|C)$) and (2) topical words (to be explained by $p(w|\theta_F)$). By fitting such a mixture model to the data, we can “factor out” the background words and obtain a discriminative topic model which would assign high probabilities to words that are common in the feedback documents but not common in the collection (thus not well explained by $p(w|C)$). The log-likelihood function is

$$\log p(F|\theta_F) = \sum_{w \in V} c(w,F) \log((1-\lambda)p(w|\theta_F) + \lambda p(w|C)),$$

where $\lambda \in [0,1]$ is a parameter indicating how much weight is put on the background model, which can be interpreted as the amount of background words we would like to factor out. The topic model θ_F can be obtained by using the ML estimator, which can be computed using the Expectation-Maximization (EM) algorithm [123].

The other approach proposed in [123] uses an idea similar to Rocchio in the vector space model [87] and assumes that θ_F is a language model that is very close to the language model of every document in the feedback document set F, but far away from the collection language model which can be regarded as an approximation of nonrelevant language model. The distance between language models is measured using KL-divergence. The problem of computing θ_F boils down to solving the following optimization problem:

$$\hat{\theta}_F = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} D(\theta||\theta_i) - \lambda D(\theta||\theta_C),$$

where θ_i is the language model estimated using document $d_i \in F$, θ_C is the background collection language model $p(w|C)$, and $\lambda \in [0,1)$ is a parameter to control the distance between the estimated θ_F and the background model θ_C. This optimization problem has an analytical solution:

$$p(w|\hat{\theta}_F) \propto \exp\left(\frac{1}{1-\lambda}\frac{1}{n}\sum_{i=1}^{n}\log p(w|\theta_i) - \frac{1}{1-\lambda}\log p(w|C)\right).$$

Both the mixture model method and the divergence minimization method are shown to be quite effective for pseudo feedback with performance comparable to or better than Rocchio [123]. However, both methods (especially divergence minimization) are also shown to be sensitive to parameter settings.

There has been some follow-up work on improving the robustness of the mixture model feedback method [107, 108]. In Tao and Zhai [107], the mixture model was extended to better integrate the original query model with the feedback documents and to allow each feedback document to potentially contribute differently to the estimated feedback topic language model. The extended model is shown to be relatively more robust than the original model, but the model is still quite sensitive to the number of documents used for feedback [107]. Moreover, due to the use of several priors, this new model has more prior parameters that need to be set manually with little guidance.

In Tao and Zhai [108], these prior parameters were eliminated through a regularized EM algorithm, and a more robust pseudo feedback model is established. Indeed, it has been shown that with no parameter tuning, the model delivers comparable performance to a well-tuned baseline pseudo feedback model.

The main ideas introduced in this new model and estimation method are the following: (1) Each feedback document is allowed to have a potentially different amount of noisy words, and the amount of noise is estimated with no need of manual tuning. This makes it more robust with respect to the number of documents used for pseudo feedback. (2) The interpolation of the original query model with the feedback model is implemented by treating the original query model as a prior in a Bayesian estimation framework. This makes the interpolation more meaningful and offers the opportunity to dynamically change the interpolation weights during the estimation process. (3) The parameter estimation process (EM algorithm) is carefully regularized so that we would start with the original query model and gradually enrich it with additional words picked up from the feedback documents. Such regularization ensures that the estimated model stays close to the original query. (4) This gradual enrichment process stops when “sufficient” new words have been picked up by the EM algorithm, where “sufficient” roughly corresponds to reaching a balance between the original query model and the new topic model picked up from the feedback documents (i.e., interpolation with a 0.5 weight).

A different approach to improving robustness of pseudo feedback is presented in Collins-Thompson and Callan [19], where the idea is to perform sampling over both the feedback documents and the query to generate alternative sets of feedback documents and alternative query variants. Feedback models obtained from each alternative set can then be combined to improve the robustness of the estimated feedback model. Experiments using a variant of the relevance model [55] as the baseline feedback method show that the proposed sampling method can improve the robustness of feedback even though not necessarily the accuracy of feedback.

5.3.2 Markov Chain Query Model Estimation

Another approach to estimating a query model is to iteratively mine the entire corpus by following a Markov chain formed by documents and terms [50]. The basic idea of this approach is to exploit term co-occurrences to learn a translation model $t(u|v)$ which can be expected to capture the semantic relations between words in the sense that $t(u|v)$ would give a high probability to word u if it is semantically related to word v. Specifically, we can imagine a surfer iteratively following a Markov chain of the form $w_0 \to d_0 \to w_1 \to d_1 \ldots$, where w_i is a word and d_i a document, and the transition probability from a document to a word is given by the document language model $p(w|d)$, while the transition probability from a word to a document is assumed to be the posterior probability $p(d|w) \propto p(w|d)p(d)$. When visiting a word, the surfer is further assumed to stop at the word with probability $1 - \alpha$ which is a parameter to be empirically set. The translation probability $t(u|v)$ can then be defined as the probability of stopping at u if the surfer starts with v. Clearly, the same Markov chain can also be exploited to compute other translation probabilities such as $t(d_1|d_2)$ or $t(d|u)$ without much modification.

Once we have such a translation model, we can assume that a user has an information need characterized by a query model θ_Q, and the user has formulated the current query Q through sampling a word from θ_Q and then "translating" it to a query word in Q according to the translation model. Given the observed Q, we can then compute the posterior probability of a word being selected from θ_Q and use this probability to estimate θ_Q:

$$p(w|\theta_Q) \propto \sum_{i=1}^{m} t(q_i|w)p(w|U),$$

where $p(w|U)$ is our prior probability that a word w would have been chosen by user U; it can be set to the collection language model $p(w|C)$ with no additional knowledge.

Intuitively, this model exploits global co-occurrences of words to expand a query and obtain an enriched query language model. However, while such a global expansion has been shown to be effective, the expansion is much more effective if the Markov chain is restricted to going through the top-ranked documents for a query [50]. Thus the method can also be regarded as a way to perform pseudo feedback with language models. The observation that local co-occurrence analysis is more effective than global co-occurrence analysis is also consistent with a study of a traditional retrieval model [116]. Intuitively, this is because the local documents (i.e., documents close to the query) can prevent noisy words from being picked from feedback due to distracting co-occurrences.

In Collins-Thompson and Callan [18], such a Markov chain expansion method has been extended to include multiple types of term associations, such as co-occurrences in an external corpus, co-occurrences in top-ranked search results, term associations obtained from an external resource (e.g., WordNet). While the expansion accuracy is not better than a strong baseline expansion method, such a massive expansion strategy is shown to be more robust.

5.3.3 Relevance Model

The work reviewed so far on using language models for IR is rooted at the query likelihood retrieval method. In 2001, another very interesting language model, called *relevance model* was developed by Lavrenko and Croft [55]. The motivation for this model comes from the difficulty in estimating model parameters in the classical probabilistic model when we do not have relevance judgments.

The classical probabilistic model can be obtained by using the same derivation as discussed in Section 3.1 and taking the document-generation decomposition of the joint probability $p(Q,D|R)$:

$$O(R|Q,D) \propto \frac{p(D|Q,R=1)}{p(D|Q,R=0)}. \tag{5.5}$$

We see that our main tasks are to estimate two document models, one for relevant documents (i.e., $p(D|Q,R=1)$) and one for nonrelevant documents (i.e., $p(D|Q,R=0)$). If we assume a multiple Bernoulli model for $p(D|Q,R)$, we will obtain precisely the Binary Independence Model proposed by Robertson and Sparck Jones [83] and further studied by others (e.g., [20, 110]). The model parameters can be estimated by using some examples of relevant and nonrelevant documents, making this an attractive model for relevance feedback.

However, when we do not have relevance judgments, it would be difficult to estimate the parameters. Croft and Harper [20] studied this problem and introduced two approximations: (1) the nonrelevant document model $p(D|Q,R=0)$ can be estimated by assuming all the documents in the collection to be nonrelevant. (2) the relevant document model $p(D|Q,R=1)$ is assumed to give a constant probability to all the query words. Using these assumptions, they showed that this classical probabilistic model would lead to a scoring formula with IDF weighting for matched terms. This is indeed a very interesting derivation and provides some probabilistic justification of IDF. However, while the first assumption is reasonable, the second is clearly an over-simplification. A more reasonable approximation may be to use some top-ranked documents as an approximation of relevant documents, i.e., follow the idea of pseudo relevance feedback. This is essentially the idea behind the relevance model work [55].

In the relevance model, a multinomial model is used to model a document, thus we can capture the term frequency naturally. (Previously, 2-Poisson mixture models had been proposed as a member of the classical probabilistic models to model term frequency, and an approximation of that model has led to the effective BM25 retrieval function [84].) Using multinomial distribution, we have

$$O(R|Q,D) \propto \frac{\prod_{i=1}^{n} p(d_i|Q,R=1)}{\prod_{i=1}^{n} p(d_i|Q,R=0)}, \tag{5.6}$$

where document $D = d_1 \cdots d_n$.

Since $p(d_i|Q,R=0)$ can be reasonably approximated by $p(d_i|C)$ (i.e., collection language model), the main challenge is to estimate $p(d_i|Q,R=1)$, which captures word occurrences in relevant documents and is called a *relevance model*. In [55], the authors proposed two methods for estimating such a relevance model, both based on the idea of using the top-ranked documents to approximate relevant documents to estimate the relevance model $p(w|Q,R=1)$.

In Model 1, they essentially use the query likelihood $p(Q|D)$ as a weight for document D and take a weighted average of the probability of word w given by each document language model. Clearly only the top ranked documents matter because other documents would have a very small or zero weight. Formally, the formula is as follows:

$$\begin{aligned} p(w|Q,R=1) &\propto p(w,Q|R=1) && (5.7)\\ &\approx \sum_{\theta_D\in\Theta} p(\theta_D)p(w|\theta_D)p(Q|\theta_D) && (5.8)\\ &= \sum_{\theta_D\in\Theta} p(\theta_D)p(w|\theta_D)\prod_{i=1}^{m} p(q_i|\theta_D), && (5.9) \end{aligned}$$

where Θ represents the set of smoothed document models in the collection. $p(\theta_D)$ can be set to uniform.

In Model 2, they compute the association between each word and the query using documents containing both query terms and the word as "bridges." The strongly associated words are then assigned high probabilities in the relevance model. Formally,

$$\begin{aligned} p(w|Q,R=1) &\propto p(Q|w,R=1)p(w) && (5.10)\\ &= p(w)\prod_{i=1}^{m}\sum_{\theta_D\in\Theta} p(q_i|\theta_D)p(\theta_D|w), && (5.11) \end{aligned}$$

where $p(\theta_D|w)$ can be computed as[2]

$$p(\theta_D|w) \propto \frac{p(w|\theta_D)p(\theta_D)}{\sum_{\theta_D\in\Theta} p(w|\theta_D)p(\theta_D)}.$$

Once again, we see that the document models that give query words high probabilities dominate the computation. Thus this model intuitively also assigns high probabilities to words that occur frequently in documents that match the query well.

[2] The formula given in [55] is $p(M_i|w) = p(w|M_i)p(w)/p(M_i)$, which is probably meant to be $p(M_i|w) = p(w|M_i)p(M_i)/p(w)$; M_i is the same as θ_D.

While both models can be potentially computed over the entire space of empirical document models, in the experiments reported in [55], the authors restricted the computation to the top 50 documents returned for each query. This not only improves the efficiency, but also improves the robustness of the estimated model as we are at a lower risk of including some distracting document models. Indeed, as shown in [55], including more documents can be potentially harmful. This is the same observation as in [50], all suggesting that these models are essentially alternative ways of implementing pseudo feedback with language models.

Both versions of the relevance model are shown to be quite effective [55]. The relevance model has also later been applied to other tasks such as cross-lingual IR [54].

Although relevance model was motivated by the classical probabilistic model, it can clearly be regarded as a way to estimate the query language model. Conceptually, there appears to be little difference between relevance model and query model, both are to model what a user is interested in. That is, we may view $p(w|Q,R=1)$ as the same as $p(w|\theta_Q)$ discussed in Section 5.2. Indeed, in some later studies [52], it was shown that scoring with the KL-divergence function works better than scoring with the classical probabilistic model for the relevance model.

5.3.4 Structured Query Models

Sometimes a query may be represented in multiple ways or semi-structured so that it has multiple fields. For example, in the case of multiple representations, one representation may be based on unigrams and another may be based on word associations extracted from some domain resources [2]. In the TREC Genomics Track, gene queries are examples of queries with multiple fields: a gene query often consists of two fields, one containing the name of a gene (usually a phrase) and one with a set of symbols [128]. In all these cases, using one single query language model to represent the query appears to an over-simplification as it does not allow us to flexibly put different weights on different representations of fields.

A common solution to these problems is to define the query model as a mixture model, which was done in [2] for combining multiple sources of knowledge about query expansion and in [128] for assigning different weights to different fields of a gene query. Specifically, let $Q = \{Q_1, \ldots, Q_k\}$ be a query with k fields or representations. The mixture query model is defined as

$$p(w|\theta_Q) = \sum_{i=1}^{k} \lambda_i p(w|\theta_{Q_i}),$$

where $p(w|\theta_{Q_i})$ is a query model corresponding to field or representation Q_i, and λ_i is the corresponding weight.

In [128], a pseudo feedback algorithm is proposed to expand each $p(w|\theta_{Q_i})$ and estimate λ_i simultaneously. The basic idea is to use each field (Q_i) to define a prior on θ_{Q_i} and fit the mixture model to a set of feedback documents in the same way as fitting the two-component mixture model for model-based feedback discussed in Section 5.3.1. Such semi-structured query model is shown to be effective.

5.4 Summary

In this section, we discussed how feedback (particularly pseudo feedback) can be performed with language models. As a generalization of query likelihood scoring, the KL-divergence retrieval model has now been established as the state-of-the-art approach for using language models to rank documents. It supports all kinds of feedback through estimating a query language model based on feedback information. We reviewed several different approaches to improving the estimation of a query language model by using word co-occurrences in the corpus. Although some approaches are meant to work on the entire corpus, they tend to work much better when restricting the estimation to using only the top-ranked documents. Thus it is fair to say that all these methods are essentially different ways to implement the traditional pseudo feedback heuristic with language models. Among all the methods, the two-component mixture model [108, 123] and the relevance model [55] appear to be most effective and robust and also are computationally feasible.

6

Language Models for Special Retrieval Tasks

Although most work on language models deals with the standard monolingual *ad hoc* search task, there has also been a lot of research on applying language models to many other special retrieval tasks, including cross-lingual retrieval, distributed IR, expert finding, personalized search, modeling redundancy, subtopic retrieval, and topic mining, among others. In this section, we will review this line of work.

6.1 Cross-lingual Information Retrieval

A major challenge in cross-lingual IR is to cross the language barrier in some way, typically involving translating either the query or the document from one language to the other. The translation model discussed in Section 4.5 can be naturally applied to solve this problem by defining the translation probability $p(u|v)$ on terms in the two different languages involved in the retrieval task. For example, u may be a word in Chinese and v a word in English. In such a case, $p(u|v)$ would give us the probability that u is a good translation of English word v in Chinese; intuitively it captures the semantic association between words in different languages.

Xu and co-authors [118] applied this idea to cross-lingual IR, and proposed the following cross-lingual query likelihood retrieval function:

$$p(Q|D) = \prod_{i=1}^{m} \left[\alpha p(q_i|C_S) + (1-\alpha) \sum_{w \in V_T} p(q_i|w) p(w|D) \right],$$

where C_S is the collection in the source language (i.e., the language of the query Q), V_T is the vocabulary set of the target language (i.e., the language of the document D), and α is a smoothing parameter.

As in the translation model for monolingual *ad hoc* retrieval, a major challenge here is to estimate the translation probability $p(q_i|w)$. In [118], the authors experimented with several options, including using a bilingual word list, parallel corpora, and a combination of them. The cross-lingual query likelihood retrieval function has been shown to be quite effective, achieving over 85% performance of monolingual retrieval baseline.

In another line of work on applying language models to CLIR, Lavrenko and co-authors [54] adapted the relevance model (Model 1) in two ways to perform CLIR, both based on the KL-divergence scoring function. The document language model θ_D is estimated in a normal way, thus it assigns probabilities to words in the target language. Their main idea is to adapt relevance model so that we can start with a query Q in the source language to estimate a query model θ_Q that can assign probabilities to words in the *target* language. This way, the query model and the document model can be compared using the KL-divergence scoring function since they are now in the same (target) language.

Their first method is to leverage a parallel corpus where documents in the source language are paired with translations of them in the target language. In this case, the document model θ_D in their relevance model can be generalized to include two separate models, one for each language. That is, $\theta_D = (\theta_D^S, \theta_D^T)$, where θ_D^S is the model for the source document and θ_D^T the model for the corresponding target document. With this setup, the relevance model can be generalized in a straightforward way to give the following probability of word w^T in the target language according to the query model θ_Q:

$$p(w^T|\theta_Q) = \sum_{\theta_D \in \Theta} p(\theta_D) p(w^T|\theta_D^T) \prod_{i=1}^{m} p(q_i|\theta_D^S).$$

The pairing of θ_D^S and θ_D^T has enabled the crossing of the language barrier.

Their second method is to leverage a bilingual dictionary to induce a translation model $p(w^S|w^T)$ and use this translation model to convert the document language model $p(w^T|D)$, which is in the target language, to a document language model for the source language $p(w^S|D)$. That is,

$$\begin{aligned} p(w^T|\theta_Q) &= \sum_{\theta_D \in \Theta} p(\theta_D) p(w^T|\theta_D) \prod_{i=1}^{m} p(q_i|\theta_D) && (6.1) \\ &= \sum_{\theta_D \in \Theta} p(\theta_D) p(w^T|\theta_D) \prod_{i=1}^{m} \sum_{w \in V^T} p(q_i|w) p(w|\theta_D). && (6.2) \end{aligned}$$

This time, the translation model $p(q_i|w)$ has enabled the crossing of the language barrier.

These models have been shown to achieve very good retrieval performance (90%–95% of a strong monolingual baseline).

6.2 Distributed Information Retrieval

Language models have also been applied to perform distributed IR. The task of distributed IR can often be decomposed into two subtasks: (1) resource selection; and (2) result fusion. Language models have been applied to both tasks.

For resource selection, which is to select the most promising collections to search based on a query, the general idea is to treat each collection as a special "document" and apply standard language models to rank collections. In an early work by Xu and Croft [117], the authors cluster the documents to form topical clusters. Each cluster is then treated as one coherent subcollection, which is then used to estimate a topic language model. The KL-divergence between the empirical query word distribution and the estimated topic language model is then used to select the most promising topical collections for further querying. Such a clustering method is shown to be effective for collection selection [117].

In [95], the authors proposed a language modeling framework for resource selection and result fusion. In this framework, documents in each collection are scored using regular query likelihood retrieval function but smoothed with the background language model corresponding to the collection. As a result, the scores of documents in different collections are strictly speaking not comparable because of the use of different background language model for smoothing. A major contribution of the work [95] is to derive an adjustment strategy that can ensure that the scores of all the documents would be comparable after adjustment.

Specifically, let D be a document in collection C_i. In general, we score the document for query Q with query likelihood and rank documents based on $p(Q|D,C_i)$. The likelihood is conditioned on C_i because of smoothing, thus directly merging results based on their query likelihood scores $p(Q|D,C_i)$ would be problematic since the scores may not be comparable. Thus they use probabilistic rules to derive a normalized form of the likelihood denoted as $p(Q|D)$, which can then be used as scores of documents for the purpose of result fusion. They show that ranking based on $p(Q|D)$ is equivalent to ranking based on $\frac{p(Q|D,C_i)}{\beta p(C_i|Q)+1}$, where β is a parameter to be empirically set. Thus when we merge the results, we just need to divide the original score $p(Q|D,C_i)$ by the normalizer $\beta p(C_i|Q) + 1$, which can be computed using Bayes' rule and the likelihood of the query given collection C_i (i.e., $p(Q|C_i)$). Their experiment results show that this language modeling approach is effective for distributed IR and outperforms a state-of-the-art method (i.e., CORI) [95].

6.3 Structured Document Retrieval and Combining Representations

Most retrieval models are designed to work on a bag of words representation of a document, but ignore any structure of a document. In reality, a document often has both intra-document structures (e.g., title vs. body) and inter-document structures (e.g., hyperlinks and topical relations), which can be potentially leveraged to improve search accuracy. This is especially true in XML retrieval and Web search. It is also common that one may obtain multiple text representations of the same document, which should be combined to improve retrieval accuracy. In all these problems, we can assume that a document D has k parts or text representations $D = \{D_1, \ldots, D_k\}$, and our goal is to rank such documents with consideration of the known structure of the document.

In Ogilvie and Callan [78], the authors have extended the basic query likelihood to address this problem. Their approach allows different parts of a document or different representations of a document to be combined with different weights. Specifically, the "generation" process of a query given a document is assumed to consist of two steps: In the first step, a part D_i is selected from the structured document D according to a selection probability $p(D_i|D)$. In the second, a query is generated using the selected part D_i. Thus, the query likelihood is given by

$$p(Q|D) = \prod_{i=1}^{m} p(q_i|D) \tag{6.3}$$

$$= \prod_{i=1}^{m} \sum_{j=1}^{k} p(q_i|D_j) p(D_j|D). \tag{6.4}$$

In [78], such a two-step generation process was not explicitly given, but their model implies such a generation process. The "part selection probability" $p(D_i|D)$ is denoted by λ_i in [78]; it can be interpreted as the weight assigned to D_i and can be set based on prior knowledge or estimated using training data. How to implement such a model efficiently was discussed in length in [77]. Experiment results show that this language modeling approach to combining multiple representations is effective. Language models have also been applied to XML retrieval by other researchers [33].

A general probabilistic propagation framework was proposed in [92] to combine probabilistic content-based retrieval models including language models with link structures (e.g., hyperlinks). Results show that the propagation framework can improve ranking accuracy over pure content-based scoring. While the framework is not specific to language models, it was shown in [92] that the performance is much better if the content-based scores can be interpreted as probabilities of relevance. Another general (nonprobabilistic) propagation framework was proposed in [81] which has been shown to be effective for improving Web search through both score propagation and term count propagation. How to integrate such propagation frameworks with language models more tightly remains an interesting future research question.

6.4 Personalized and Context-sensitive Search

In personalized search, we would like to use more user information to better infer a user's information need. With language models, this means we would like to estimate a better query language model with more user information. In [94, 105], the authors proposed several estimation methods for estimating a query language model based on implicit feedback information, including the previous queries and clickthroughs of a user. These methods are shown to be effective for improving search accuracy for a new related query.

In [94], implicit feedback within a single search session is considered. This is to simulate a scenario when the initial search results were not satisfactory to the user, so the user would reformulate the query potentially multiple times. The feedback information available consists of the previous queries and the snippets of viewed documents (i.e., clickthrough information). Given the user's current query, the question is how to use such feedback information to improve the estimate of the query language model θ_Q. In [94], four different methods were proposed to solve this problem, all essentially leading to some interpolation of many unigram language models estimated using different feedback information, respectively. Different methods mainly differ in the way to assign weights to different types of information (e.g., queries vs. snippets). Experiment results show that using the history information, especially the snippets of viewed documents, can improve search accuracy for the current query. It is also shown to be beneficial to use a dynamic interpolation coefficient similar to Dirichlet prior smoothing.

In [105], implicit feedback using the entire search history of a user is considered. Since in this setup, there is potentially noisy information in the search history, it is important to filter out such noise when estimating the query language model. The idea presented in [105] for solving this problem is the following: First, each past query is treated as a unit and represented by the snippets of the top-ranked search results. Second, the search results (snippets) of the current query are used to assign a weight to each past query based on the similarity between the search results of the past query and those of the current one. The weighting helps filter out any noisy information in the history. Finally, the query language model is estimated as a weighted combination of unigram language models for each past query. The second step is implemented by using a mixture model with each past query contributing a component language model to fit the current search results. The EM algorithm is used to compute the ML estimate so that we can obtain optimal weights for all the past queries. Intuitively, the weight on each past query indicates how well the search results of that query can explain the current search results, i.e., similarity between that past query and the current query. Evaluation shows that such a language modeling approach to query estimation based on search history can improve performance substantially [105].

6.5 Expert Finding

The task of expert finding as set up in the TREC Enterprise Track is the following: Given a list of names and emails of candidate experts and text collections where their expertise may be mentioned, retrieve experts with expertise on a given topic (described by keywords). Language models have been applied to this task with reasonable success [3, 25].

In [25], a general probabilistic model is presented for solving this problem with an analogous derivation to the one given in [51]. Specifically, three random variables are introduced: (1) T for topic; (2) C for a candidate expert; (3) $R \in \{0,1\}$ for relevance. The goal is to rank the candidates according to the conditional probability $p(R = 1|T,C)$. Following the derivation in [51], the authors derived two families of models corresponding to two different ways of factoring the joint probability $p(T,C|R)$, either as $p(T|R,C)p(C|R)$, which is called *topic generation* model or $p(C|T,R)p(T|R)$, which is called *candidate generation* model. They also proposed three techniques to improve the estimation of models: (1) a mixture model for modeling the candidate mentions, which can effectively assign different weights to different representations of an expert; (2) topic expansion to enrich topic representation; (3) email-based candidate prior to prefer candidates with many email mentions. These techniques are shown to be empirically effective.

In [3], the authors proposed two different topic generation models for expert finding. In both models (indeed, also in most other studies of expert finding with the TREC Enterprise Track setup), the documents where mentions of a candidate and terms describing a topic co-occur serve as “bridges” to assess the associations between a candidate and a topic. However, the two models differ in the way a topic is “generated” from a candidate. In Model 1, a topic is generated by generating each word in the topic *independently*, thus the generation of two words of the topic can potentially be going through a different document, and a bridge document only needs to match the candidate and *one* topic word well. In Model 2, the whole topic is generated together using the same document as a bridge, thus requiring the bridge document to match both the candidate and the *entire* topic well. Thus intuitively Model 2 appears to more reasonable than Model 1. Indeed, empirical experiments also show that Model 2 outperforms Model 1 [3]. This is a case where some analysis of the assumptions made in designing language models can help assess the effectiveness of a model for retrieval.

6.6 Modeling Redundancy and Novelty

A basic task in many applications is to measure the redundancy between two documents; the purpose is often to remove or reduce the redundancy in the documents. Language models can be used to naturally solve this problem.

For example, in [121, 122], a simple two-component mixture model is used to measure the redundancy (or equivalently novelty) of a document D_2 with respect to another document D_1. The idea is to assume that the redundancy of D_1 with respect to D_2 corresponds to how well we can predict the content of D_1 using a language model estimated based on D_2. Intuitively, if D_1 is very similar to D_2, then we should expect the model based on D_2 to predict D_1 well (i.e., give high probability to D_1), whereas if they are quite different, the model would not predict D_1 well.

Formally, let θ_{D_2} be a language model estimated using D_2, we define the redundancy of D_1 with respect to D_2 as

$$\lambda^* = \arg\max_{\lambda} \log p(D_1|\lambda, \theta_{D_2}) \tag{6.5}$$

$$= \arg\max_{\lambda} \sum_{w \in V} c(w, D_1) \log(\lambda p(w|\theta_{D_2}) + (1 - \lambda) p(w|C)), \tag{6.6}$$

where $p(w|C)$ is a background collection language model.

Essentially, this is to let the background model and θ_{D_2} to compete for explaining D_1, and $\lambda^* \in [0,1]$ indicates the relative "competitiveness" of θ_{D_2} to the background model, thus intuitively captures the redundancy. λ^* can be computed using the EM algorithm. The novelty can be defined as $1 - \lambda^*$.

A similar but slightly more sophisticated three-component mixture model was proposed in [131] in order to capture novelty in information filtering.

Note that the redundancy/novelty captured in this way is asymmetric in the sense that if we switch the roles of D_1 and D_2, the redundancy value would be different. This is reasonable as in general, redundancy is asymmetric (considering a case where one document is part of another document). Another way of measuring redundancy/novelty with language models is to compute the cross-entropy between two document language models to obtain asymmetric similarities [49].

6.7 Predicting Query Difficulty

Yet another use of language models is to predict query difficulty [21]. The idea is to compare the query model and the collection language model and a query would be assumed to be difficult if its query model is close to the collection language model. The assumption made here is that a discriminative query tends to be easier and the discriminativeness of a query can be measured by the KL-divergence of the query model and the collection model.

Specifically, a measure called "query clarity" is defined in [21] as follows:

$$clarity(Q) = \sum_{w \in V} p(w|\theta_Q) \log \frac{p(w|\theta_Q)}{p(w|C)},$$

where θ_Q is usually an expanded query model using any feedback-based query model estimation method (e.g., mixture model [123] or relevance model [55]). Positive correlation between the clarity scores and retrieval accuracy has been observed [21].

6.8 Subtopic Retrieval

The subtopic retrieval task represents an interesting retrieval task because it requires modeling the *dependency* of relevance of individual documents [122]. Given a topic query, the task of subtopic retrieval is to retrieve documents that can cover as many subtopics of the topic as possible. If we are to solve the problem with a traditional retrieval model, we likely would have a great deal of redundancy in the top ranked documents. As a result, although most top-ranked documents may be relevant to the query, they may all cover just one subtopic, thus we do not end up having a high coverage of subtopics.

Intuitively, we may solve this problem by attempting to remove the redundancy in the search results, hoping that by avoiding covering already covered subtopics, we will have a higher chance of covering new subtopics quickly. This is precisely the idea explored in [122], where the authors used the novelty measure discussed in Section 6.6 in combination with the query likelihood relevance scoring to iteratively select the best document that is both relevant and different from the already picked documents, a strategy often called maximal marginal relevance (MMR) ranking [16].

In [121], topic models (PLSA [38] and LDA [9]) (to be discussed in more detail in Section 6.9) are applied to model the underlying subtopics and a KL-divergence retrieval function is then applied to rank documents based on subtopic representation. This method has not worked as well as the MMR method reported in [122], but it may be possible to combine such a subtopic representation with word-level representation to improve the performance.

6.9 Topic Mining

The probabilistic latent semantic analysis (PLSA) model was introduced by Hofmann in [37, 38] as a model for analyzing and extracting the latent topics in text documents. In [38], Hofmann has shown that using the latent topics discovered by PLSA to represent documents can improve retrieval performance (called probabilistic latent semantic indexing, or PLSI). Later, many different extensions of this model have been proposed, mostly for the purpose of mining (extracting) latent topics in text and revealing interesting topical patterns (e.g., temporal topical trends).

The basic idea of PLSA is to assume that each word in a document is generated from a finite mixture model with k multinomial component models (i.e., k unigram language models). Formally, let D be a document and $\theta_1, \ldots, \theta_k$ be k multinomial distributions over words, representing k latent topics. Associated with D we have a document-specific topic selection probability distribution $p_D(i)$ $(i = 1, \ldots, k)$, which indicates the probability of selecting θ_i to generate a word in the document. The log-likelihood of document D is then

$$\log p(D) = \sum_{w \in V} c(w, D) \log \left(\sum_{i=1}^{k} p_D(i) p(w|\theta_i) \right),$$

where V is the vocabulary set, $c(w,D)$ is the count of word w in D. PLSA can be estimated using the standard maximum likelihood estimator (with the Expectation-Maximization (EM) algorithm [22]) to obtain parameter values for θ_i and $p_D(i)$. Clearly, if $k = 1$, PLSA degenerates to the simple unigram language model. PLSA is generally used to fit a *set* of documents. Since the topic models θ_i are tied across the documents, they can capture clusters of words that co-occur with each other, and help discover interesting latent topics in a collection of text.

There are two problems with PLSA: (1) it is not really a generative model because the topic selection probability is defined in a document-specific way; (2) it has many parameters, making it hard to find a global maximum in parameter estimation. To address these limitations, Blei and co-authors [9] proposed Latent Dirichlet Allocation (LDA) as an extension of PLSA. The main idea is to define $p_D(i)$ in a "generative" way by drawing the distribution $p_D(i)$ from a Dirichlet distribution. This not only gives us a generative model that can be used to sample "future documents," but also reduces the number of parameters significantly. However, the estimation of PLSA is no longer as simple as using the standard EM algorithm, and tends to be computationally much more expensive [9, 71]. Many extensions of LDA have since been proposed to model coordinated data, hierarchical topics, and temporal topic patterns (see e.g., [7, 8, 10, 57, 103]).

Although LDA is advantageous over PLSA as a generative model, for the purpose of mining topics, it is unclear whether regularizing the topic choices with a parametric Dirichlet distribution is advantageous; intuitively, this makes the estimated $p_D(i)$ less discriminative. PLSA has also been extended in several studies mostly to accommodate a topic hierarchy [36], incorporate context variables such as time and location [66], and analyze sentiments [65]. In [130], a background topic is introduced to PLSA to make the extracted topic models more focusing on the content words rather than the common words in the collection.

6.10 Summary

In this section, we reviewed a wide spectrum of work on using language models for different kinds of special retrieval tasks. Since the central topic of this review paper is ad hoc retrieval, we have intentionally focused on applications of language models in *unsupervised* settings and left out work on using language models in *supervised* learning settings (where labeled training data is needed) because the latter, which includes tasks such as text categorization, information filtering, and topic tracking and detection, is better reviewed through comparing the generative language models with many other competing supervised learning methods, notably discriminative models.

7

Unifying Different Language Models

In addition to the study of individual retrieval models using language modeling, there has also been some work attempting to establish a general formal framework to unify different language models and facilitate *systematic* explorations of language models in information retrieval. We have seen from previous sections that with language models we may rank documents using three different strategies: (1) query likelihood (i.e., computing $p(Q|D)$ or $p(Q|\theta_D)$); (2) document likelihood ratio (i.e., computing $p(D|Q,R=1)/p(D|Q,R=0)$); and (3) KL-divergence (i.e., computing $D(\theta_Q||\theta_D)$). The first two can be derived from the same ranking criterion $p(R=1|Q,D)$ as shown in [51], thus they naturally follow the probability ranking principle [85]. An interesting question is whether we can further unify the third one (i.e., KL-divergence scoring) with the first two. One major advantage of unifying these different scoring methods is that we will obtain a general retrieval framework that can serve as a road map to explore variations of language models systematically.

7.1 Risk Minimization

A major work in this line is the risk minimization framework [50, 121, 127]. The basic idea of this framework is to formalize the retrieval problem generally as a decision problem with Bayesian decision theory, which provides a solid theoretical foundation for thinking about problems of action and inference under uncertainty [6]. Language models are introduced into the framework as models for the observed data, particularly the documents and queries.

Specifically, we assume that we observe the user $\mathcal{U}$, the query $\mathbf{q}$, the document source $\mathcal{S}$, and the collection of documents $\mathcal{C}$. We view a query as being the output of some probabilistic process associated with the user $\mathcal{U}$, and similarly, we view a document as being the output of some probabilistic process associated with an author or document source $\mathcal{S}$. A query (document) is the result of choosing a model, and then generating the query (document) using that model.

The system is supposed to choose an optimal action to take in response to these observations. An *action* corresponds to a possible response of the system to a query. We can represent all actions by $\mathcal{A} = \{(D_i, \pi_i)\}$, where $D_i \subseteq \mathcal{C}$ is a subset of $\mathcal{C}$ (results) and $\pi_i \in \Pi$ is some presentation strategy.

In Bayesian decision theory, to each such action $a_i = (D_i, \pi_i) \in \mathcal{A}$ there is associated a *loss* $L(a_i, \theta, F(\mathcal{U}), F(\mathcal{S}))$, which in general depends upon all of the parameters of our model, $\theta \equiv (\theta_Q, \{\theta_i\}_{i=1}^N)$ as well as any relevant user factors $F(\mathcal{U})$ and document source factors $F(\mathcal{S})$.

In this framework, the *expected risk of action* a_i is given by

$$R(D_i, \pi_i \,|\, \mathcal{U}, \mathbf{q}, \mathcal{S}, \mathcal{C}) = \int_\Theta L(D_i, \pi_i, \theta, F(\mathcal{U}), F(\mathcal{S}))\, p(\theta \,|\, \mathcal{U}, \mathbf{q}, \mathcal{S}, \mathcal{C})\, d\theta,$$

where the *posterior distribution* is given by

$$p(\theta \,|\, \mathcal{U}, \mathbf{q}, \mathcal{S}, \mathcal{C}) \propto p(\theta_Q \,|\, \mathbf{q}, \mathcal{U}) \prod_{i=1}^{N} p(\theta_i \,|\, \mathbf{d}_i, \mathcal{S}).$$

The Bayes decision rule is then to choose the action $\mathbf{a}^*$ with the least expected risk:

$$\mathbf{a}^* = (D^*, \pi^*) = \arg\min_{D,\pi} R(D, \pi \,|\, \mathcal{U}, \mathbf{q}, \mathcal{S}, \mathcal{C}).$$

That is, to select D^* and present D^* with strategy π^*.

The risk minimization framework provides a general way to frame a retrieval problem using language models. With different instantiations of the query model θ_Q and document models θ_i, it can accommodate potentially many different language models, while the loss function can be instantiated in different ways to reflect different retrieval/ranking criteria. It has been shown in [121, 127] that the risk minimization framework covers many existing retrieval models as special cases and can serve as a roadmap for exploring new models. For example, the probability ranking principle (thus both the query likelihood scoring method and the document likelihood ratio scoring method) can be derived by making the independent loss assumption[1] and defining the loss function based on the relevance status of a document. The KL-divergence scoring function can be justified by defining the loss of returning a document based on the KL-divergence of its language model and the query language model.

In general, we do not have to make the independent loss assumption and can define the loss function over the *entire ranked list* of documents. Thus in the risk minimization framework, it is possible to go beyond independent relevance to capture redundancy between documents (e.g., as done in [122]), which is hard to capture when the retrieval problem is formulated as computing a score based on matching one query with *one* document. See Section 6.8 for more discussion of using language models to solve the subtopic retrieval problem.

The optimization setup of the risk minimization is quite general and offers potential for combining language models with the line of work on learning to rank (e.g., [12, 43] and their recent extensions [13, 15]).

7.2 Generative Relevance

Another important work is the generative relevance framework developed in Lavrenko's thesis [52]. In [52], the following generative relevance

[1] The independent loss assumption says that the loss of returning a document in response to a query is independent of returning other documents. This assumption clearly does not hold in reality because redundant documents do not have independent loss. Thus the probability ranking principle has the limitation of not being able to model redundancy among the search results.

hypothesis was proposed:

Generative Relevance Hypothesis: For a given information need, queries expressing that need and documents relevant to that need can be viewed as independent random samples from the same underlying generative model.

Lavrenko developed three different retrieval functions under this hypothesis (i.e., query likelihood, document likelihood, and KL-divergence) and proposed a general technique called kernel-based allocation for estimating various kinds of language models [52]. The generative relevance hypothesis has two important implications from the perspective of deriving retrieval models: (1) It naturally accommodates matching of queries and documents even if they are in different languages (as in the case of cross-lingual retrieval) or in different media (e.g., text queries on images). (2) It makes it possible to estimate and improve a relevant *document* language model based on examples of *queries* and vice versa. Conceptually, the generative relevance framework can be regarded as a special case of risk minimization when document models and query models are assumed to be in the same space.

8

Summary and Outlook

8.1 Language Models vs. Traditional Retrieval Models

It has been a long-standing challenge in IR research to develop robust and effective retrieval models. As a new generation of probabilistic retrieval models, language modeling approaches have several advantages over traditional retrieval models such as the vector-space model and the classical probabilistic retrieval model:

First, these language models generally have a good statistical foundation. This makes it possible to leverage many established statistical estimation methods to set parameters in a retrieval function as demonstrated in [108, 125]. Following rigorous statistical modeling also forces any assumptions to be made explicit. A good understanding of such assumptions often helps diagnose the weakness and strength of a model and thus better interpret experiment results.

Second, they provide a principled way to address the critical issue of text representation and term weighting. The issue of term weighting has long been recognized as critical, but before language modeling approaches were proposed, this issue had been traditionally addressed mostly in a heuristic way. Language models, multinomial unigram language models particular, can incorporate term frequencies and document length normalization naturally into a probabilistic model. While such connection has also been made in the classic probabilistic retrieval model (e.g., [83]), the estimation of parameters was not addressed as seriously as in the language models.

Third, language models can often be more easily adapted to model various kinds of complex and special retrieval problems than traditional models as discussed in Section 6. The benefit has largely come from the availability of many well-understood statistical models such as finite mixture models, which can often be estimated easily by using the EM algorithm.

However, the language modeling approaches also have some deficiencies as compared with traditional models:

First, there is a lack of *explicit* discrimination in most of the language models developed so far. For example, in the query likelihood retrieval function, the IDF effect is achieved through smoothing the document language model with a background model. While this can be explained by modeling the noise in the query, it seems to be a rather unnatural way to penalize matching common words, at least as compared with the traditional TF-IDF weighting. Such a lack of discrimination is indeed a general problem with all generative models as they are designed to describe what the data looks like rather than how the data differs.

Second, the language models have been found to be less robust than the traditional TF-IDF model in some cases and can perform poorly or be very sensitive to parameter setting. For example, the feedback methods proposed in [123] are shown to be sensitive to parameter setting, whereas a traditional method such as Rocchio appears to be more robust. This may be the reason why language models have not yet been able to outperform well-tuned full-fledged traditional methods consistently and convincingly in TREC evaluation. In particular, BM25 term weighting coupled with Rocchio feedback remains a strong baseline which is at least as competitive as any language modeling approach for many tasks.

Third, some sophisticated language models can be computationally expensive (e.g., the translation model), which may limit their uses in large-scale retrieval applications.

It should be noted that these relative advantages and disadvantages are based on a quite vague distinction between language models and classical probabilistic retrieval models. Indeed, the boundary is not very clear. Conceptually, any probabilistic model of text can be called a language model. In this sense, the classical probabilistic retrieval model such as the Robertson–Sparck Jones model is certainly also a language model (i.e., multiple Bernoulli). However, for historical reasons, the term language model tends to be used to refer to either the use of a multinomial model (or a higher order n-gram model such as bigram or trigram language model) or the query likelihood retrieval function and its generalization KL-divergence retrieval function. Thus readers should be careful about the vague distinction of the so-called language models from other (traditional) probabilistic models.

8.2 Summary of Research Progress

Since the pioneering work by Ponte and Croft [80], a lot of progress has been made in studying the language modeling approaches to IR, which we briefly reviewed in this survey. The following is an incomplete list of some of the most important developments:

- Framework and justification for using LMs for IR have been established: The query likelihood retrieval method has been shown to be a well-justified model according to the probability ranking principle [51]. General frameworks such as the risk minimization framework [50, 121, 127] and the generative relevance framework [52] offer road maps for systematically applying language models to retrieval problems.
- Many effective models have been developed and they often work well for multiple tasks:

  — The KL-divergence retrieval model [50, 52, 123], which covers the query likelihood retrieval model, has been found to be a solid and empirically effective retrieval model that can easily incorporate many advanced language models; many methods have been

  developed to improve estimation of query language models.
  - Dirichlet prior smoothing has been recognized as an effective smoothing method for retrieval [124]. The KL-divergence retrieval model combined with Dirichlet prior smoothing represents the state-of-the-art baseline method for the language modeling approaches to IR.
  - The translation model proposed in [4] enables handling polysemy and synonyms in a principled way with a great potential for supporting semantic information retrieval.
  - The relevance model [52, 55] offers an elegant solution to the estimation problem in the classical probabilistic retrieval model as well as serves as an effective feedback method for the KL-divergence retrieval model.
  - Mixture unigram language models have been shown to be very powerful and can be useful for many purposes such as pseudo feedback [123], improving model discriminativeness [35], and modeling redundancy [122, 131].
- It has been shown that completely automatic tuning of parameters is possible for both nonfeedback retrieval [125] and pseudo feedback [108].
- LMs can be applied to virtually any retrieval task with great potential for modeling complex IR problems (as surveyed in Section 6).

For practitioners who want to apply language models in specific applications, the KL-divergence retrieval function combined with Dirichlet prior smoothing for estimating document language models and either relevance model or mixture model for estimating query language models can be highly recommended. These methods have all been implemented in the Lemur toolkit (http://www.lemurproject.org/).

8.3 Future Directions

Despite much progress has been made in applying language models to IR, there are still many challenges to be solved to fully develop the potential of such models. The following is a list of some interesting opportunities for future research:

Challenge 1: Develop an efficient, robust and effective language model for ad hoc retrieval that can (1) optimize retrieval parameters automatically, (2) perform as well as or better than well-tuned traditional retrieval methods with pseudo feedback (e.g., BM25 with Rocchio), and (3) be computed as efficiently as traditional retrieval methods. Would some kind of language model eventually replace the currently popular BM25 and Rocchio? How to implement IDF more explicitly in a language modeling approach may be an important issue to further study. Relaxing the assumption that the same words occur independently in a document (e.g., by using the Dirichlet Compound Model) may also be necessary to capture TF normalization more accurately.

Challenge 2: Demonstrate consistent and substantial improvement by going beyond unigram language models. While there has been some effort in this direction, the empirical performance improvement of the more sophisticated models over the simple models tends to be insignificant. This is consistent with what has been observed in traditional retrieval models. Would we ever be able to achieve significant improvement over the unigram language models by using higher-order n-gram models or capturing limited syntactic/semantic dependencies among words? As we go beyond unigram language models, reliable estimation of the model becomes more challenging due to the problem of data sparseness. Thus developing better estimation techniques (e.g., those that can lead to optimal weighting of phrases conditioned on weighting of single words) may be critical for making more progress in this direction.

Challenge 3: Develop language models to support personalized search. Using more user information and a user's search context to better infer a user's information need is essential for optimizing search accuracy. This is especially important when the search results are not satisfactory and the user would reformulate the query many times. How can we use language models to accurately represent a user's interest and further incorporate such knowledge into a retrieval model? Detailed analysis of user actions (e.g., skipping some results and viewing others, deleting query terms but adding them back later, recurring interests vs. *adhoc* information needs) may be necessary to obtain an accurate representation of a user's information need.

Challenge 4: Develop language models that can support "life-time learning." One important advantage of language models is the potential benefit from improved estimation of the models based on additional training data. As a search engine is being used, we will be able to collect a lot of implicit feedback information such as clickthroughs. How can we develop language models that can learn from all such feedback information from all the users to optimize retrieval results for future queries? From the viewpoint of personalized search, how can we leverage many users of a system to improve performance for a particular user (i.e., supporting collaborative search)? Translation models appear to be especially promising in this direction, and they are complementary with the recently developed discriminative models for learning to rank documents such as RankNet [12] and Ranking SVM [43]. It should be extremely interesting to study how to combine these two complementary approaches.

Challenge 5: Develop language models that can model document structures and subtopics. Most existing work on studying retrieval models, including work on language models, has assumed a simple bag-of-words representation of text. While such a representation ensures that the model would work for any text, in a specific application, documents often have certain structures that can be potentially exploited to improve search accuracy. For example, often it is some part of a long document that is relevant. How can we model potentially different subtopics in a single document and match only the relevant part of a document with a query? Mixture models and hidden Markov models may be promising in this direction.

Challenge 6: Generalize language models to support ranking of both unstructured and structured data. Traditionally, structured data and unstructured data (text) have been managed in different ways with structured data mainly handled through a relational database while unstructured data through an information retrieval system, leading to two different research communities (i.e., the database community and the information retrieval community). Recently, however, the boundary between the two communities seems to become vague. First, as exploratory search on databases becomes more and more popular on the Web, DB researchers are now paying much attention to the problem of ranking structured data in a database. The information needs to be satisfied are very similar to those in a retrieval system. Second, some database fields may contain long text (e.g., abstracts of research surveys), while most text documents also have some structured meta-data (e.g., authors, dates). Thus a very interesting question is whether we can generalize language models to develop unified probabilistic models for searching/ranking both structured data and unstructured data. The INEX initiative (http://inex.is.informatik.uni-duisburg.de/) has stimulated a lot of research in developing XML retrieval models (i.e., semi-structured data retrieval models), but we are still far from a unified model for unstructured, semi-structured, and structured data.

Challenge 7: Develop language models for hypertext retrieval. As an abstract representation, the Web can be regarded a hypertext collection. Language models developed so far have not explicitly incorporated hyperlinks and the associated anchor text into the model. How can we use language modeling to develop a hypertext retrieval model for Web search? How should we define a generative model for hypertext?

Challenge 8: Develop/Extend language models for retrieval with complex information needs. Language models are natural for modeling topical relevance. But in many retrieval applications, a user's information need consists of multiple dimensions of preferences with topical relevance being only one of them. Other factors such as readability, genre, and sentiment may also be important. How can we use language models to capture such nontopical aspects? How can we develop or extend language models to optimize ranking of documents based on multiple factors? In this direction, recent work has shown that the learning-to-rank approaches are quite promising, thus again it would be very interesting to study how to combine the language modeling approaches (generative approaches) with the learning-to-rank approaches (discriminative approaches).

Acknowledgments

I want to thank Jamie Callan, editor of Foundations and Trends in IR, for the opportunity to write this review, for his technical feedback about the review, and for his periodical reminders which ensured progress in my writing. I also want to thank him for encouraging me to give a tutorial on this topic at several conferences. This survey is largely a written elaboration of the tutorial. I am very grateful to John Lafferty for supervising my dissertation on this topic. The conceptual framework and a substantial part of this survey are based on my dissertation work. I want to thank Donald Metzler and two anonymous reviewers for their many useful comments and suggestions on how to improve this survey. (Naturally, any errors that remain are solely my own responsibility.) I also wants to thank many peer researchers, especially W. Bruce Croft, Stephen Robertson, Victor Lavrenko, Rong Jin, Tao Tao, David A. Evans, and Wessel Kraaij, for their useful discussions on various issues about the topic covered in this survey. I give special thanks to James Finlay and Mike Casey of Now Publishers for their help and support in preparing the final version of this review.

My own research work covered in this survey is supported in part by the Advanced Research and Development Activity in Information

Technology (ARDA) under its Statistical Language Modeling for Information Retrieval Research Program, by the National Science Foundation under a CAREER grant (IIS-0347933), by Google, Microsoft, and IBM through their research programs, and by the Alfred P. Sloan Foundation through a research fellowship. Any opinions, findings, and conclusions, or recommendations expressed in this paper are those of the author and do not necessarily reflect the views of the sponsors.

References

[1] G. Amati and C. J. V. Rijsbergen, "Probabilistic models of information retrieval based on measuring the divergence from randomness," *ACM Transactions on Information System*, vol. 20, pp. 357–389, 2002.

[2] J. Bai, J.-Y. Nie, G. Cao, and H. Bouchard, "Using query contexts in information retrieval," in *Proceedings of ACM SIGIR 2007*, pp. 15–22, 2007.

[3] K. Balog, L. Azzopardi, and M. de Rijke, "Formal models for expert finding in enterprise corpora," in *Proceedings of SIGIR-06*, 2006.

[4] A. Berger and J. Lafferty, "Information retrieval as statistical translation," in *Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 222–229, 1999.

[5] A. L. Berger and J. D. Lafferty, "The Weaver system for document retrieval," in *Proceedings of TREC 1999*, 1999.

[6] J. Berger, *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlap, 1985.

[7] D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum, "Hierarchical topic models and the nested Chinese restaurant process," in *Neural Information Processing Systems (NIPS) 16*, 2003.

[8] D. Blei and J. Lafferty, "Correlated topic models," in *Proceedings of NIPS '05: Advances in Neural Information Processing Systems 18*, 2005.

[9] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[10] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 113–120, 2006.

[11] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "A statistical approach to

machine translation," *Computational Linguistics*, vol. 16, no. 2, pp. 79–85, 1990.

[12] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *ICML '05: Proceedings of the 22nd International Conference on Machine Learning*, pp. 89–96, USA, New York, NY: ACM, 2005.

[13] C. J. C. Burges, R. Ragno, and Q. V. Le, "Learning to rank with nonsmooth cost functions," in *Proceedings of NIPS 2006*, (B. Scholkopf, J. C. Platt, and T. Hoffman, eds.), pp. 193–200, 2006.

[14] G. Cao, J.-Y. Nie, and J. Bai, "Integrating word relationships into language models," in *Proceedings of the 2005 ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 298–305, 2005.

[15] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: From pairwise approach to listwise approach," in *Proceedings of ICML 2007, Volume 227 of ACM International Conference Proceeding Sereies*, (Z. Ghahramani, ed.), pp. 129–136, 2007.

[16] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of SIGIR'98*, pp. 335–336, 1998.

[17] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Technical Report TR-10-98, Harvard University, 1998.

[18] K. Collins-Thompson and J. Callan, "Query expansion using random walk models," in *Proceedings of ACM CIKM 2005*, pp. 704–711, 2005.

[19] K. Collins-Thompson and J. Callan, "Estimation and use of uncertainty in pseudo-relevance feedback," in *Proceedings of ACM SIGIR 2007*, pp. 303–310, 2007.

[20] W. B. Croft and D. Harper, "Using probabilistic models of document retrieval without relevance information," *Journal of Documentation*, vol. 35, pp. 285–295, 1979.

[21] S. Cronen-Townsend, Y. Zhou, and W. B. Croft, "Predicting query performance," in *Proceedings of the 25th Annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2002)*, pp. 299–306, August 2002.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of Royal Statistical Society B*, vol. 39, pp. 1–38, 1977.

[23] D. A. Evans and C. Zhai, "Noun-phrase analysis in unrestricted text for information retrieval," in *Proceedings of ACL 1996*, pp. 17–24, 1996.

[24] H. Fang, T. Tao, and C. Zhai, "A formal study of information retrieval heuristics," in *Proceedings of the 2004 ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 49–56, 2004.

[25] H. Fang and C. Zhai, "Probabilistic models for expert finding," in *Proceedings of ECIR 2007*, pp. 418–430, 2007.

[26] N. Fuhr, "Probabilistic models in information retrieval," *The Computer Journal*, vol. 35, pp. 243–255, 1992.

[27] N. Fuhr, “Language models and uncertain inference in information retrieval,” in *Proceedings of the Language Modeling and IR Workshop*, pp. 6–11, May 31–June 1 2001.

[28] J. Gao, J.-Y. Nie, G. Wu, and G. Cao, “Dependence language model for information retrieval,” in *SIGIR ’04: Proceedings of the 27th Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 170–177, USA, New York, NY: ACM, 2004.

[29] D. Grossman and O. Frieder, *Information retrieval: Algorithms and heuristics.* Springer, 2004.

[30] D. Hiemstra, “A probabilistic justification for using tf x idf term weighting in information retrieval,” *International Journal on Digital Libraries*, vol. 3, pp. 131–139, 2000.

[31] D. Hiemstra, “Using language models for information retrieval,” PhD Thesis, University of Twente, 2001.

[32] D. Hiemstra, “Term-specific smoothing for the language modeling approach to information retrieval: The importance of a query term,” in *Proceedings of ACM SIGIR 2002*, pp. 35–41, 2002.

[33] D. Hiemstra, “Statistical language models for intelligent XML retrieval,” in *Intelligent Search on XML Data*, pp. 107–118, 2003.

[34] D. Hiemstra and W. Kraaij, “Twenty-One at TREC-7: Ad-hoc and cross-language track,” in *Proceedings of Seventh Text REtrieval Conference (TREC-7)*, pp. 227–238, 1998.

[35] D. Hiemstra, S. Robertson, and H. Zaragoza, “Parsimonious language models for information retrieval,” in *SIGIR ’04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 178–185, USA, New York, NY: ACM, 2004.

[36] T. Hofmann, “The cluster-abstraction model: Unsupervised learning of topic hierarchies from text data,” in *Proceedings of IJCAI’ 99*, pp. 682–687, 1999.

[37] T. Hofmann, “Probabilistic latent semantic analysis,” in *Proceedings of UAI 1999*, pp. 289–296, 1999.

[38] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of ACM SIGIR’99*, pp. 50–57, 1999.

[39] F. Jelinek, *Statistical Methods for Speech Recognition.* MIT Press, 1997.

[40] F. Jelinek and R. Mercer, “Interpolated estimation of Markov source parameters from sparse data,” in *Pattern Recognition in Practice*, (E. S. Gelsema and L. N. Kanal, eds.), pp. 381–402, 1980.

[41] H. Jin, R. Schwartz, S. Sista, and F. Walls, “Topic tracking for radio, tv broadcast, and newswire,” in *Proceedings of the DARPA Broadcast News Workshop*, pp. 199–204, 1999.

[42] R. Jin, A. G. Hauptmann, and C. Zhai, “Title language model for information retrieval,” in *Proceedings of ACM SIGIR 2002*, pp. 42–48, 2002.

[43] T. Joachims, “Optimizing search engines using clickthrough data,” in *Proceedings of the ACM KDD 2002*, pp. 133–142, 2002.

[44] 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*, vol. ASSP-35, pp. 400–401, 1987.

[45] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 181–184, 1995.

[46] W. Kraaij, "Variations on language modeling for information retrieval," PhD Thesis, University of Twente, 2004.

[47] W. Kraaij, T. Westerveld, and D. Hiemstra, "The importance of prior probabilities for entry page search," in *Proceedings of ACM SIGIR 2002*, pp. 27–34, 2002.

[48] O. Kurland and L. Lee, "Corpus structure, language models, and *ad hoc* information retrieval," in *SIGIR '04: Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval*, pp. 194–201, ACM Press, 2004.

[49] O. Kurland and L. Lee, "PageRank without hyperlinks: Structural re-ranking using links induced by language models," in *SIGIR '05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 306–313, USA, New York, NY: ACM, 2005.

[50] J. Lafferty and C. Zhai, "Document language models, query models, and risk minimization for information retrieval," in *Proceedings of SIGIR'01*, pp. 111–119, September 2001.

[51] J. Lafferty and C. Zhai, "Probabilistic relevance models based on document and query generation," in *Language Modeling and Information Retrieval*, (W. B. Croft and J. Lafferty, eds.), pp. 1–6, Kluwer Academic Publishers, 2003.

[52] V. Lavrenko, "A generative theory of relevance," PhD Thesis, University of Massachusetts, Amherst, 2004.

[53] V. Lavrenko, J. Allan, E. DeGuzman, D. LaFlamme, V. Pollard, and S. Thomas, "Relevance models for topic detection and tracking," in *Proceedings of the Second International Conference on Human Language Technology Research*, pp. 115–121, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.

[54] V. Lavrenko, M. Choquette, and W. B. Croft, "Cross-lingual relevance models," in *Proceedings of ACM SIGIR 2002*, pp. 175–182, 2002.

[55] V. Lavrenko and W. B. Croft, "Relevance-based Language Models," in *Proceedings of SIGIR'01*, pp. 120–127, September 2001.

[56] D. D. Lewis, "Representation and learning in information retrieval," Technical Report 91-93, University of Massachusetts, 1992.

[57] W. Li and A. McCallum, "Pachinko allocation: DAG-structured mixture models of topic correlations," in *ICML '06: Proceedings of the 23rd International Conference on Machine Learning*, pp. 577–584, 2006.

[58] X. Li and W. B. Croft, "Time-based language models," in *CIKM '03: Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pp. 469–475, USA, New York, NY: ACM, 2003.

[59] X. Liu and W. B. Croft, "Passage retrieval based on language models," in *CIKM '02: Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pp. 375–382, USA, New York, NY: ACM, 2002.

[60] X. Liu and W. B. Croft, "Cluster-based retrieval using language models," in *SIGIR '04: Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval*, pp. 186–193, ACM Press, 2004.

[61] D. MacKay and L. Peto, "A hierarchical Dirichlet language model," *Natural Language Engineering*, vol. 1, pp. 289–307, 1995.

[62] R. E. Madsen, D. Kauchak, and C. Elkan, "Modeling word burstiness using the Dirichlet distribution," in *ICML '05: Proceedings of the 22nd International Conference on Machine Learning*, pp. 545–552, USA, New York, NY: ACM, 2005.

[63] C. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[64] A. McCallum and K. Nigam, "A comparison of event models for Naive Bayes text classification," in *AAAI-1998 Learning for Text Categorization Workshop*, pp. 41–48, 1998.

[65] Q. Mei, H. Fang, , and C. Zhai, "A study of Poisson query generation model for information retrieval," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 319–326, 2007.

[66] Q. Mei and C. Zhai, "A mixture model for contextual text mining," in *Proceedings of KDD '06*, pp. 649–655, 2006.

[67] Q. Mei, D. Zhang, and C. Zhai, "A general optimization framework for smoothing language models on graph structures," in *SIGIR '08: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 611–618, USA, New York, NY: ACM, 2008.

[68] D. Metzler and W. B. Croft, "A Markov random field model for term dependencies," in *Proceedings of the 2005 ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 472–479, 2005.

[69] D. Metzler, V. Lavrenko, and W. B. Croft, "Formal multiple-Bernoulli models for language modeling," in *SIGIR '04: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 540–541, USA, New York, NY: ACM, 2004.

[70] D. H. Miller, T. Leek, and R. Schwartz, "A hidden Markov model information retrieval system," in *Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 214–221, 1999.

[71] T. Minka and J. Lafferty, "Expectation-propagation for the generative aspect model," in *Proceedings of the UAI 2002*, pp. 352–359, 2002.

[72] R. Nallapati and J. Allan, "Capturing term dependencies using a language model based on sentence trees," in *CIKM '02: Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pp. 383–390, USA, New York, NY: ACM, 2002.

[73] R. Nallapati, B. Croft, and J. Allan, "Relevant query feedback in statistical language modeling," in *CIKM '03: Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pp. 560–563, USA, New York, NY: ACM, 2003.

[74] H. Ney, U. Essen, and R. Kneser, "On structuring probabilistic dependencies in stochastic language modeling," *Computer Speech and Language*, vol. 8, pp. 1–38, 1994.

[75] K. Ng, "A maximum likelihood ratio information retrieval model," in *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, (E. Voorhees and D. Harman, eds.), pp. 483–492, 2000.

[76] P. Ogilvie and J. Callan, "Experiments using the Lemur toolkit," in *Proceedings of the 2001 TREC conference*, 2002.

[77] P. Ogilvie and J. Callan, "Using language models for flat text queries in XML retrieval," in *Proceedings of the Initiative for the Evaluation of XML Retrieval Workshop (INEX 2003)*, 2003.

[78] P. Ogilvie and J. P. Callan, "Combining document representations for known-item search," in *Proceedings of ACM SIGIR 2003*, pp. 143–150, 2003.

[79] J. Ponte, "A language modeling approach to information retrieval," PhD Thesis, University of Massachusetts at Amherst, 1998.

[80] J. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proceedings of the ACM SIGIR'98*, pp. 275–281, 1998.

[81] T. Qin, T.-Y. Liu, X.-D. Zhang, Z. Chen, and W.-Y. Ma, "A study of relevance propagation for web search," in *Proceedings of SIGIR 2005*, pp. 408–415, 2005.

[82] L. R. Rabiner, "A tutorial on hidden Markov models," *Proceedings of the IEEE*, vol. 77, pp. 257–285, 1989.

[83] S. Robertson and K. Sparck Jones, "Relevance weighting of search terms," *Journal of the American Society for Information Science*, vol. 27, pp. 129–146, 1976.

[84] S. Robertson and S. Walker, "Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval," in *Proceedings of SIGIR'94*, pp. 232–241, 1994.

[85] S. E. Robertson, "The probability ranking principle in IR," *Journal of Documentation*, vol. 33, pp. 294–304, December 1977.

[86] S. E. Robertson, S. Walker, K. Sparck Jones, M. M. Hancock-Beaulieu, and M. Gatford, "Okapi at TREC-3," in *The Third Text Retrieval Conference (TREC-3)*, (D. K. Harman, ed.), pp. 109–126, 1995.

[87] J. Rocchio, "Relevance feedback in information retrieval," in *The SMART Retrieval System: Experiments in Automatic Document Processing*, pp. 313–323, Prentice-Hall Inc., 1971.

[88] T. Roelleke and J. Wang, "A parallel derivation of probabilistic information retrieval models," in *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 107–114, New York, NY, USA: ACM, 2006.

[89] G. Salton, *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer.* Addison-Wesley, 1989.

[90] G. Salton and M. McGill, *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[91] G. Salton, C. S. Yang, and C. T. Yu, "A theory of term importance in automatic text analysis," *Journal of the American Society for Information Science*, vol. 26, pp. 33–44, Jan–Feb 1975.

[92] A. Shakery and C. Zhai, "A probabilistic relevance propagation model for hypertext retrieval," in *Proceedings of CIKM 2006*, pp. 550–558, 2006.

[93] A. Shakery and C. Zhai, "Smoothing document language models with probabilistic term count propagation," *Information Retrieval*, vol. 11, pp. 139–164, 2008.

[94] X. Shen, B. Tan, and C. Zhai, "Context-sensitive information retrieval using implicit feedback," in *Proceedings of SIGIR 2005*, pp. 43–50, 2005.

[95] L. Si, R. Jin, J. P. Callan, and P. Ogilvie, "A language modeling framework for resource selection and results merging," in *Proceedings of CIKM 2002*, pp. 391–397, 2002.

[96] A. Singhal, C. Buckley, and M. Mitra, "Pivoted document length normalization," in *Proceedings of the 1996 ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 21–29, 1996.

[97] F. Song and W. B. Croft, "A general language model for information retrieval," in *Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 279–280, 1999.

[98] K. Sparck Jones, S. Robertson, D. Hiemstra, and H. Zaragoza, "Language modeling and relevance," in *Language Modeling for Information Retrieval*, (W. B. Croft and J. Lafferty, eds.), pp. 57–72, 2003.

[99] M. Spitters and W. Kraaij, "Language models for topic tracking," in *Language Modeling for Information Retrieval*, pp. 95–124, 2003.

[100] M. Srikanth and R. Srihari, "Biterm language models for document retrieval," in *SIGIR '02: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 425–426, USA, New York, NY: ACM, 2002.

[101] M. Srikanth and R. Srihari, "Exploiting syntactic structure of queries in a language modeling approach to IR," in *CIKM '03: Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pp. 476–483, USA, New York, NY: ACM, 2003.

[102] M. Srikanth and R. Srihari, "Incorporating query term dependencies in language models for document retrieval," in *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp. 405–406, USA, New York, NY: ACM, 2003.

[103] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths, "Probabilistic author-topic models for information discovery," in *Proceedings of KDD'04*, pp. 306–315, 2004.

[104] T. Strzalkowski and B. Vauthey, "Information retrieval using robust natural language processing," in *Proceedings of the 30th annual meeting on Association for Computational Linguistics*, pp. 104–111, Morristown, NJ, USA: Association for Computational Linguistics, 1992.

[105] B. Tan, X. Shen, and C. Zhai, "Mining long-term search history to improve search accuracy," in *KDD*, pp. 718–723, 2006.

[106] T. Tao, X. Wang, Q. Mei, and C. Zhai, "Language model information retrieval with document expansion," in *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pp. 407–414, Morristown, NJ, USA: Association for Computational Linguistics, 2006.

[107] T. Tao and C. Zhai, "Mixture clustering model for pseudo feedback in information retrieval," in *Proceedings of the 2004 Meeting of the International Federation of Classification Societies*, Spriner, 2004.

[108] T. Tao and C. Zhai, "Regularized estimation of mixture models for robust pseudo-relevance feedback," in *Proceedings of ACM SIGIR 2006*, pp. 162–169, 2006.

[109] H. Turtle and W. B. Croft, "Evaluation of an inference network-based retrieval model," *ACM Transactions on Information Systems*, vol. 9, pp. 187–222, July 1991.

[110] C. J. van Rijbergen, "A theoretical basis for the use of co-occurrence data in information retrieval," *Journal of Documentation*, pp. 106–119, 1977.

[111] C. J. van Rijsbergen, "A non-classical logic for information retrieval," *The Computer Journal*, vol. 29, no. 6, pp. 481–485, 1986.

[112] X. Wang, H. Fang, and C. Zhai, "Improve retrieval accuracy for difficult queries using negative feedback," in *CIKM '07: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, pp. 991–994, USA, New York, NY: ACM, 2007.

[113] X. Wei and W. Bruce Croft, "LDA-based document models for ad-hoc retrieval," in *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 178–185, USA, New York, NY: ACM, 2006.

[114] S. K. M. Wong and Y. Y. Yao, "A probability distribution model for information retrieval," *Information Processing and Management*, vol. 25, pp. 39–53, 1989.

[115] S. K. M. Wong and Y. Y. Yao, "On modeling information retrieval with probabilistic inference," *ACM Transactions on Information Systems*, vol. 13, pp. 69–99, 1995.

[116] J. Xu and W. B. Croft, "Query expansion using local and global document analysis," in *Proceedings of the SIGIR'96*, pp. 4–11, 1996.

[117] J. Xu and W. B. Croft, "Cluster-based language models for distributed retrieval," in *Proceedings of the SIGIR'99*, pp. 254–261, 1999.

[118] J. Xu, R. Weischedel, and C. Nguyen, "Evaluating a probabilistic model for cross-lingual information retrieval," in *SIGIR '01: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 105–110, USA, New York, NY: ACM, 2001.

[119] H. Zaragoza, D. Hiemstra, and M. E. Tipping, "Bayesian extension to the language model for ad hoc information retrieval," in *Proceedings of ACM SIGIR 2003*, pp. 4–9, 2003.

[120] C. Zhai, "Fast statistical parsing of noun phrases for document indexing," in *5th Conference on Applied Natural Language Processing (ANLP-97)*, pp. 312–319, March 31–April 3 1997.

[121] C. Zhai, "Risk minimization and language modeling in text retrieval," PhD Thesis, Carnegie Mellon University, 2002.

[122] C. Zhai, W. W. Cohen, and J. Lafferty, "Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval," in *Proceedings of ACM SIGIR'03*, pp. 10–17, August 2003.

[123] C. Zhai and J. Lafferty, "Model-based feedback in the language modeling approach to information retrieval," in *Tenth International Conference on Information and Knowledge Management (CIKM 2001)*, pp. 403–410, 2001.

[124] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," in *Proceedings of ACM SIGIR'01*, pp. 334–342, September 2001.

[125] C. Zhai and J. Lafferty, "Two-stage language models for information retrieval," in *Proceedings of ACM SIGIR'02*, pp. 49–56, August 2002.

[126] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Transactions on Information Systems*, vol. 2, pp. 179–214, 2004.

[127] C. Zhai and J. Lafferty, "A risk minimization framework for information retrieval," *Information Processing Management*, vol. 42, pp. 31–55, 2006.

[128] C. Zhai, X. Lu, X. Ling, A. Velivelli, X. Wang, H. Fang, and A. Shakery, "UIUC/MUSC at TREC 2005 Genomics Track," in *Proceedings of TREC 2005*, 2005.

[129] C. Zhai, T. Tao, H. Fang, and Z. Shang, "Improving the robustness of language models — UIUC TREC 2003 robust and genomics experiments," in *Proceedings of TREC 2003*, pp. 667–672, 2003.

[130] C. Zhai, A. Velivelli, and B. Yu, "A cross-collection mixture model for comparative text minning," in *Proceeding of the 10th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 743–748, 2004.

[131] Y. Zhang, J. Callan, and T. Minka, "Redundancy detection in adaptive filtering," in *Proceedings of SIGIR'02*, pp. 81–88, August 2002.

[132] Y. Zhou and W. B. Croft, "Document quality models for web ad hoc retrieval," in *CIKM '05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pp. 331–332, USA, New York, NY: ACM, 2005.