Risk Minimization and Language Modeling in Text Retrieval – Thesis Summary

ChengXiang Zhai
Language Technologies Institute
School of Computer Science
Carnegie Mellon University

July 21, 2002

Abstract

This thesis presents a new general probabilistic framework for text retrieval based on Bayesian decision theory. In this framework, queries and documents are modeled using statistical language models, user preferences are modeled through loss functions, and retrieval is cast as a risk minimization problem. This risk minimization framework not only unifies several existing retrieval models within one general probabilistic framework, but also facilitates the development of new principled approaches to text retrieval through the use of statistical language models. We explore three interesting special cases of the framework. In the case of a two-stage language modeling approach, we show that it is possible to achieve excellent retrieval performance without any ad hoc parameter tuning by exploiting statistical estimation methods to set the retrieval parameters completely automatically. In another case of a KL-divergence retrieval model, we demonstrate that it is possible to improve retrieval performance by using improved language models estimated based on feedback documents. Finally, in the case of non-traditional aspect retrieval models, we show that it is possible to use language models to capture redundancy and sub-topics in documents, and to perform "context-sensitive" ranking of documents based on both relevance and novelty of documents.

1 Introduction

Recent years have seen an explosive growth of the volume of information. Information retrieval is one of the most useful techniques to address the problem of information overload. The retrieval of textual information (i.e., text retrieval) is especially important, because the most frequently wanted information is often textual, and techniques for retrieving textual information can be useful for retrieving other media information when companion text is available.

The task of text retrieval can be defined as, taking as input, a document collection (i.e., a set of unordered text documents) and a user query (i.e., a description of the user's information need), to identify a subset of documents that can satisfy the user's information need. Since it is generally very hard, if not impossible, for a user to prescribe the exact information need completely and precisely with a query, text retrieval is really an "ill-formulated" task, in the sense that the correctness of the solution to the retrieval problem can only be evaluated by a user empirically. Indeed, the criterion for judging whether a particular set of documents would satisfy the user's information need, or the notion of "*relevance*", is inherently impossible to formalize, as it is generally imprecise and depends on the situation or context of the retrieval task.

Two assumptions are often made to simplify the retrieval task: (1) *Independent relevance*. The relevance of a document is assumed to be independent of other documents, including those already retrieved. (2) *Topical relevance*. The relevance of a document is assumed to mean the level of *topical* relevance of the document with respect to the query. Under these two constraints, the retrieval task is essentially to evaluate the *topical* relevance value of each document *independently* with respect to a query, which makes the retrieval task more tractable. Both assumptions have been made in most traditional retrieval models, even though none of them can be expected to hold in reality.

Over the decades, many different retrieval models have been proposed, studied, and tested. Their mathematical basis spans a large spectrum, including algebra, logic, probability and statistics. The existing models can be roughly grouped into three major categories, depending on how they define/measure relevance. In the first category, relevance is measured by the *similarity* between a query and a document. The *vector space model* is the most well-known model of this type, in which a document and a query are represented as two term vectors in a high-dimensional term space, and the similarity between the query and the document is typically measured by the dot product of the two vectors or the cosine of the angle formed by the two vectors (Salton et al., 1975; Salton and McGill, 1983; Salton, 1989). In the second category, a binary random variable is used to model relevance and probabilistic models are used to estimate the value of this *relevance variable*. Different models of this category mainly differ in their way of defining the probabilistic model and estimating the probability of relevance. Most *classical probabilistic retrieval models* belong to this category(Maron and Kuhns, 1960; Robertson and Sparck Jones, 1976; van Rijsbergen, 1979; Robertson et al., 1981; Fuhr, 1992). The *language modeling approach* proposed recently has also been shown to be a special case of a general probabilistic relevance model(Lafferty and Zhai, 2001b). In the third category, relevance is measured by the *uncertainty* in inferring queries from documents or vise versa. Examples of this category include the *logic-based* probabilistic inference model(van Rijsbergen, 1986; Wong and Yao, 1995), the *inference network* model(Turtle and Croft, 1991), and the *spread-activation model*(Kwok, 1995).

Although a lot of progress has been made in the field of text retrieval, especially the improving of empirical performance on large document collections due to the annual TREC evaluation workshop(Voorhees and Harman, 2001), the integration of theory and practice in text retrieval has so far been quite weak. Theoretical guidance and formal principles

have rarely led to good performance directly; a lot of heuristic parameter tuning must be used in order to achieve satisfactory performance. This is evident in the large number of parameter-tuning experiments reported in virtually every paper published in the TREC proceedings (Voorhees and Harman, 2001). It is thus a significant scientific challenge to develop principled retrieval approaches that also perform well empirically.

This thesis attempts to address this fundamental challenge. It presents a new general probabilistic framework for text retrieval based on Bayesian decision theory. In this framework, queries and documents are modeled using *statistical language models*, user preferences are modeled through *loss functions*, and retrieval is cast as a *risk minimization* problem. This risk minimization framework not only unifies several existing retrieval models within one general probabilistic framework, but also facilitates the development of new principled approaches to text retrieval through the use of statistical language models. Three interesting special cases of this framework are further explored in the thesis:

1. Two-stage language models

   While traditional retrieval models rely heavily on *ad hoc* parameter tuning to achieve satisfactory retrieval performance, the use of language models in the risk minimization framework makes it possible to exploit statistical estimation methods to improve retrieval performance and set retrieval parameters *automatically*. As a special case of the framework, we present a two-stage language model that, according to extensive evaluation, achieves excellent retrieval performance without any ad hoc parameter tuning.

2. Kullback-Leibler divergence retrieval models

   Using language models in retrieval also makes it possible to improve retrieval performance through using improved language models and estimation methods. As another special case of the risk minimization framework, we derive a Kullback-Leibler divergence retrieval model that can exploit *feedback documents* to improve the estimation of query models. Feedback has so far been dealt with heuristically in the language modeling approach to retrieval. The KL-divergence model provides a more natural way of performing feedback by treating it as *query model updating*. We propose two specific query model updating algorithms based on feedback documents. Evaluation indicates that both algorithms are effective for feedback.

3. Aspect retrieval models

   The risk minimization retrieval framework further allows for incorporating user factors beyond the traditional notion of topical relevance. We present a family of *non-traditional* retrieval models that are appropriate for the aspect retrieval task. Specifically, we present language models that can capture *redundancy* and *sub-topics* in documents, and study loss functions that can rank documents in terms of *both relevance and sub-topic diversity*. Evaluation shows that the proposed language models can effectively capture redundancy and can outperform the relevance-based ranking method for the aspect retrieval task.

In the following sections, we first give an overview of the risk minimization framework, and then summarize our exploration of its three special cases.

2 The Risk Minimization Retrieval Framework

2.1 The basic formulation

In general, a retrieval system can be regarded as an interactive information service system that answers a user's query by presenting a list of documents. After seeing the presented documents, the user may reformulate a query, which is then executed by the system to produce another new list of documents to present. The cycle goes on like this. At each cycle, the retrieval system needs to choose a subset of documents and present them to the user in some way, based on the information available to the system, which includes the current user, the user's query, the source of documents, and a specific document collection. The retrieval problem can thus be viewed as a *decision* problem for the system.

The basic idea of the risk minimization retrieval framework is to formalize such a general retrieval decision problem with Bayesian decision theory, which provides a solid theoretical foundation for thinking about problems of action and inference under uncertainty(Berger, 1985).

In terms of Bayesian decision theory, our *observations* are 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. A set of documents is the result of generating each document independently, possibly from a different model. (The independence assumption is not essential, and is made here only to simplify the presentation.) The query model could, in principle, encode detailed knowledge about a user's information need and the context in which they make their query. Similarly, the document model could encode complex information about a document and its source or author.

More formally, let θ_Q denote the parameters of a query model, and let θ_D denote the parameters of a document model. A user $\mathcal{U}$ generates a query by first selecting θ_Q, according to a distribution $p(\theta_Q \,|\, \mathcal{U})$. Using this model, a query $\mathbf{q}$ is then generated with probability $p(\mathbf{q} \,|\, \theta_Q)$. Similarly, the source selects a document model θ_D according to a distribution $p(\theta_D \,|\, \mathcal{S})$, and then uses this model to generate a document $\mathbf{d}$ according to $p(\mathbf{d} \,|\, \theta_D)$. Thus, we have Markov chains $\mathcal{U} \rightarrow \theta_Q \rightarrow \mathbf{q}$ and $\mathcal{S} \rightarrow \theta_D \rightarrow \mathbf{d}$. This is illustrated in Figure 1

Let $\mathcal{C} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$ be a collection of documents obtained from source $\mathcal{S}$. We denote by θ_i the model that generates document $\mathbf{d}_i$. Our observations are thus $\mathcal{U}$, $\mathbf{q}$, $\mathcal{S}$, and $\mathcal{C}$.

An *action* corresponds to a possible response of the system to a query. For example, one can imagine that the system would return an *unordered subset* of documents to the user. Alternatively, a system may decide a ranking of documents and present a *ranked list*

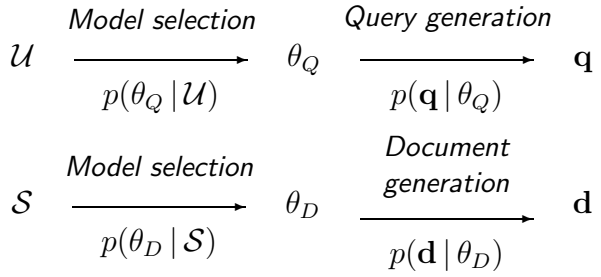

Figure 1: Generative model of query $\mathbf{q}$ and document $\mathbf{d}$.

of documents. Yet another possibility is to cluster the (relevant) documents and present a *structured view* of documents.

Generally, we can think of a retrieval action as a *compound decision* involving *selecting* a subset of documents D from $\mathcal{C}$ and *presenting* them to the user who has issued query $\mathbf{q}$ according to some presentation strategy π. Let Π be the set of all possible presentation strategies. 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^*.

This is the basic formulation of retrieval as a decision problem in the risk minimization framework, which involves searching for D^* and π^* simultaneously. In principle, strategy can be fairly arbitrary. Practically, however, we need to be able to quantify the loss associated with a presentation strategy.

2.2 Special cases

Interesting special cases of the risk minimization framework can be obtained by considering a specific loss function and specific document/query language models.

When the loss function does *not* depend on the presentation strategy, that is, all we care about is to select an optimal subset of documents for presentation, the risk minimization framework leads to the following general *set-based retrieval model*.

$$D^* \quad = \quad \arg\min_D \int_\Theta L(D, \theta, F(\mathcal{U}), F(\mathcal{S}))\, p(\theta \,|\, \mathcal{U}, \mathbf{q}, \mathcal{S}, \mathcal{C})\, d\theta$$

The traditional *Boolean retrieval model* can be viewed as a special case of this general set-based retrieval framework, where we have no uncertainty about the query models and document models (e.g., $\theta_Q = \mathbf{q}$ and $\theta_i = \mathbf{d}_i$), and the following loss function is used:

$$L(D, \theta, F(\mathcal{U}), F(\mathcal{S})) = \sum_{\mathbf{d} \in D} -\delta(\mathbf{d}, \mathbf{q})$$

where $\delta(\mathbf{d}, \mathbf{q}) = 1$ if and only document $\mathbf{d}$ satisfies the Boolean query $\mathbf{q}$; otherwise $\delta(\mathbf{d}, \mathbf{q}) = -1$. This loss function is actually quite general, in the sense that if we allow $\delta(\mathbf{d}, \mathbf{q})$ to be any *deterministic* retrieval rule applied to query $\mathbf{q}$ and document $\mathbf{d}$, such that, $\delta(\mathbf{d}, \mathbf{q}) > 0$ if $\mathbf{d}$ is relevant to $\mathbf{q}$, otherwise $\delta(\mathbf{d}, \mathbf{q}) < 0$, then, the loss function would always result in a retrieval strategy that involves making an *independent* binary retrieval decision for each document according to δ. There are many other possibilities to specialize the set-based retrieval method, but exploring them is not the main focus of this thesis.

If we assume that the loss function does *not* depend on the selected subset of documents, and our presentation strategy corresponds to a ranking of *all* the documents in the collection, we can obtain a general ranking-based retrieval model.

Let us denote an action by π_i, which is a permutation over $[1..N]$, i.e., a complete ordering of the N documents in the collection $\mathcal{C}$. In order to characterize the loss associated with a *ranking* of documents, we assume a "sequential browsing model" of the user – the user would read the documents in the order and stop wherever is appropriate. Thus, the *actual* loss (or equivalently utility) of a ranking would depend on where the user actually stops. That is, the utility is affected by the user's browsing behavior, which we could model through a probability distribution over all the ranks that a user might stop at. Formally, let s_i denote the probability that the user would stop reading after seeing the top i documents. We have $\sum_{i=1}^{N} s_i = 1$. We can treat $s_1, ..., s_N$ as user factors given by $F(\mathcal{U})$.

Given this setup, we can now define the loss for a ranking as the *expected* loss under the assumed "stopping rank" distribution.

$$L(\pi, \theta, F(\mathcal{U}), F(\mathcal{S})) = \sum_{i=1}^{N} s_i l(\pi(1:i), \theta, F(\mathcal{U}), F(\mathcal{S}))$$

where, $l(\pi(1:i), \theta, F(\mathcal{U}), F(\mathcal{S}))$ is the actual loss that would be incurred if the user actually views the first i documents according to π. Note that $L(\pi, \theta, F(\mathcal{U}), F(\mathcal{S}))$ and

$l(\pi, \theta, F(\mathcal{U}), F(\mathcal{S}))$ are different: the former is the *expected* loss of the ranking under the user's "stopping probability distribution", while the latter is the *exact* loss of the ranking when the user actually views the whole list.

Assuming that the user would view the documents in the order of being presented, and the total loss of viewing i documents is the sum of the loss associated with viewing each individual document, we have the following reasonable *additive* decomposition of the loss:

$$l(\pi(1:i), \theta, F(\mathcal{U}), F(\mathcal{S})) = \sum_{j=1}^{i} l(d_{\pi^j} | d_{\pi^1}, ..., d_{\pi^{j-1}}, \theta, F(\mathcal{U}), F(\mathcal{S}))$$

where $l(d_{\pi^j} | d_{\pi^1}, ..., d_{\pi^{j-1}}, \theta, F(\mathcal{U}), F(\mathcal{S}))$ is the *conditional* loss of viewing d_{π^j} given that the user has already viewed $(d_{\pi^1}, ..., d_{\pi^{j-1}})$.

Putting all these together, the optimal ranking is given by

$$\pi^* = \arg\min_{\pi} \sum_{i=1}^{N} s_i \sum_{j=1}^{i} \int_{\Theta} l(d_{\pi^j} | d_{\pi^1}, ..., d_{\pi^{j-1}}, \theta, F(\mathcal{U}), F(\mathcal{S})) p(\theta | \mathbf{q}, \mathcal{C}, \mathcal{U}, \mathcal{S}) d\theta$$

Define the following *conditional* risk

$$r(\mathbf{d}_k | \mathbf{d}_1, ..., \mathbf{d}_{k-1}, \mathbf{q}, \mathcal{C}, \mathcal{U}, \mathcal{S}) \stackrel{\text{def}}{=} \int_{\Theta} l(\mathbf{d}_k | \mathbf{d}_1, ..., \mathbf{d}_{k-1}, \theta, F(\mathcal{U}), F(\mathcal{S})) p(\theta | \mathbf{q}, \mathcal{C}, \mathcal{U}, \mathcal{S}) d\theta$$

which can be interpreted as the expected risk of user's viewing document $\mathbf{d}_k$ given that $\mathbf{d}_1, ..., \mathbf{d}_{k-1}$ have already been previously viewed. We can write

$$\begin{aligned} R(\pi | \mathbf{q}, \mathcal{C}, \mathcal{U}, \mathcal{S}) &= \sum_{i=1}^{N} s_i \sum_{j=1}^{i} r(d_{\pi^j} | d_{\pi^1}, ..., d_{\pi^{j-1}}, \mathbf{q}, \mathcal{C}, \mathcal{U}, \mathcal{S}) \\ &= \sum_{j=1}^{N} (\sum_{i=j}^{N} s_i) r(d_{\pi^j} | d_{\pi^1}, ..., d_{\pi^{j-1}}, \mathbf{q}, \mathcal{C}, \mathcal{U}, \mathcal{S}) \end{aligned}$$

This is the general framework for *ranking* documents within the risk minimization framework. It basically says that the optimal ranking is the one minimizing the expected conditional loss (under stopping distribution) associated with sequentially viewing each document in the order given by the ranking.

There are two further special cases of this general ranking model, corresponding to the use of an *independent* and a *dependent* loss function, respectively. With an independent loss function, we assume that the loss of viewing each document is independent of viewing others. In this case, the optimal ranking π^* can be shown to be *independent* of $\{s_i\}$, and is just ranking documents according to the *individual* risk of each document, given by

$$r(\mathbf{d} | \mathbf{q}, \mathcal{C}, \mathcal{U}, \mathcal{S}) = \int_{\Theta_Q} \int_{\Theta_D} l(\mathbf{d}, \theta_Q, \theta_D, F(\mathcal{U}), F(\mathcal{S})) p(\theta_Q \,|\, \mathbf{q}, \mathcal{U}) p(\theta_D \,|\, \mathbf{d}, \mathcal{S}) \, d\theta_D \, d\theta_Q$$

This is equivalent to the situation when we assume each possible action is to present a *single* document. The loss function $l(\mathbf{d}, \theta_Q, \theta_D, F(\mathcal{U}), F(\mathcal{S}))$ can be interpreted as the loss associated with presenting/viewing document $\mathbf{d}$.

Such a general independent loss ranking model is a generalization of the *Probability Ranking Principle* proposed in (Robertson, 1977), and can be shown to cover several existing retrieval models as special cases, including the probabilistic relevance model and the language modeling approach proposed recently(Lafferty and Zhai, 2001a).

Independent loss is not really a realistic assumption; the loss of viewing one document generally depends on the documents already viewed. For example, if the user has already seen the same document or a similar document, then, it should incur a much greater loss than if the document is completely new to the user. When the independence assumption does not hold, the complexity of finding the optimal ranking makes the computation intractable. One practical solution is to use a *greedy algorithm* to construct a sub-optimal ranking. Specifically, we will "grow" the target ranking by choosing the document at each rank, starting from the very first rank. Suppose we already have a partially constructed ranking $\pi(1:i)$, we would choose a document that *minimizes the risk increase* for rank $i+1$. Let k be a possible index of document to be considered for rank $i+1$, and $\pi(1:i,k)$ represent the ordering $(d_{\pi(1:i)^1}, ..., d_{\pi(1:i)^i}, d_k)$. Then, the increase of risk for picking d_k at rank $i+1$ is

$$\begin{aligned}\delta(k|\pi(1:i)) &= R(\pi(1:i,k)|\mathbf{q},\mathcal{C},\mathcal{U},\mathcal{S}) - R(\pi(1:i)|\mathbf{q},\mathcal{C},\mathcal{U},\mathcal{S}) \\ &= s_{i+1}(r(d_k|d_{\pi(1:i)^1}, ..., d_{\pi(1:i)^i}, \mathbf{q},\mathcal{C},\mathcal{U},\mathcal{S}) + \\ &\quad \sum_{j=1}^{i} r(d_j|d_{\pi(1:i)^1}, ..., d_{\pi(1:i)^{j-1}}, \mathbf{q},\mathcal{C},\mathcal{U},\mathcal{S}))\end{aligned}$$

Thus, to extend $\pi(1:i)$, we should choose

$$\begin{aligned}k^* &= \arg\min_k \delta(k|\pi(1:i)) \\ &= \arg\min_k r(d_k|d_{\pi(1:i)^1}, ..., d_{\pi(1:i)^i}, \mathbf{q},\mathcal{C},\mathcal{U},\mathcal{S})\end{aligned}$$

That is, at each step, we just need to evaluate

$$\delta'(k|\pi(1:i)) = r(d_k|d_{\pi(1:i)^1}, ..., d_{\pi(1:i)^i}, \mathbf{q},\mathcal{C},\mathcal{U},\mathcal{S})$$

and choose the k that minimizes $\delta'(k|\pi(1:i))$.

This gives us a general greedy and context-dependent ranking algorithm. With this algorithm, again, we see that the "optimal" ranking does not depend on the stopping probabilities s_i. In Section 5, we will discuss several special cases of this algorithm, including the *Maximal Marginal Relevance method* (MMR)(Carbonell and Goldstein, 1998).

3 Two-stage Language Models

In the special case of using an independent loss function for ranking, let us consider the following special loss function, indexed by a small constant ϵ,

$$l_\epsilon(\mathbf{d}, \theta_Q, \theta_D, F(\mathcal{U}), F(\mathcal{S})) \;=\; \begin{cases} 0 & \text{if } \Delta(\theta_Q, \theta_D) \leq \epsilon \\ c & \text{otherwise} \end{cases}$$

where $\Delta : \Theta_Q \times \Theta_D \rightarrow \mathbb{R}$ is a model distance function, and c is a constant positive cost. Thus, the loss is zero when the query model and the document model are close to each other, and is c otherwise.

Using this loss function, with some approximations, we will be essentially ranking documents according to the posterior probability that the user used the estimated document model as the query model, which is given by

$$p(\hat{\theta}_D \,|\, \mathbf{q}, \mathcal{U}) \;\propto\; p(\mathbf{q} \,|\, \hat{\theta}_D, \mathcal{U}) p(\hat{\theta}_D \,|\, \mathcal{U})$$

This is the basic two-stage language model retrieval formula. This formula has the following interpretation: $p(\mathbf{q} \,|\, \hat{\theta}_D, \mathcal{U})$ captures how well the estimated document model $\hat{\theta}_D$ explains the query, whereas $p(\hat{\theta}_D \,|\, \mathcal{U})$ encodes our prior belief that the user would use $\hat{\theta}_D$ as the query model. While this prior could be exploited to model different document sources or other document characteristics, we assume a uniform prior in this thesis.

The name "*two-stage*" comes from the fact that to score a document using this formula, we would *first* estimate a document language model, and *then* compute the query likelihood using a query generative model which is based on the estimated document model. From the viewpoint of *smoothing*, we can regard such a two-stage language modeling approach as involving a two-stage smoothing of the original document model. The first-stage smoothing happens when we estimate the document language model, and the second-stage is implemented through the query generative model.

One special case of this two-stage language modeling approach is the original language modeling approach proposed in (Ponte and Croft, 1998), which can be obtained by using the simplest unigram language model as the query generation model – essentially we would have only the first-stage smoothing. In this case, it can be shown that if the document language model $\hat{\theta}_D$ is smoothed with the *collection language model*, the retrieval function would, in effect, perform a term weighting that is similar to *TF-IDF weighting* with document *length normalization*. In general, smoothing is very important for estimating an accurate language model, and we have found that the retrieval performance can be very *sensitive* to smoothing. We have done many experiments with several smoothing methods, and have found that *Dirichlet prior* smoothing method is effective for first-stage smoothing. However, we have also found that the first-stage alone is *inadequate* – we must also perform the second-stage smoothing to model any possible non-informative words in the query(Zhai and Lafferty, 2001b).

A more interesting special case involves using a simple *mixture model* for query generation, which results in a second-stage smoothing with linear interpolation (simplified Jelinek-

Mercer). The overall effect is the following *two-stage smoothing* retrieval formula:

$$p(\mathbf{q} \mid \hat{\theta}_D, \lambda, \mathcal{U}) = \prod_{i=1}^{m} \left((1-\lambda) \frac{c(q_i, d) + \mu\, p(q_i \mid \mathcal{S})}{|\mathbf{d}| + \mu} + \lambda\, p(q_i \mid \mathcal{U}) \right)$$

where, μ is the (first-stage) Dirichlet prior smoothing parameter and λ is the (second-stage) Jelinek-Mercer interpolation parameter. $p(q_i \mid \mathcal{S})$ is the collection language model, and $p(. \mid \mathcal{U})$ is a *query background language model*, which can explain the non-informative words in the query. This two-stage smoothing method is shown to reveal a more regular sensitivity pattern of smoothing empirically.

We propose a *leave-one-out* method for estimating the first-stage Dirichlet prior parameter and a *mixture model* for estimating the second-stage interpolation parameter. These methods allow us to set the retrieval parameters automatically, yet adaptively according to different databases and queries. Evaluation on five different databases and four types of queries indicates that the two-stage smoothing method with the proposed parameter estimation scheme consistently gives retrieval performance that is close to, or better than, the best results attainable using a single smoothing method, achievable only through an exhaustive parameter search. The effectiveness and robustness of the two-stage smoothing approach, along with the fact that there is no ad hoc parameter tuning involved, make it a solid baseline approach for evaluating retrieval models(Zhai and Lafferty, 2002).

4 KL-divergence Retrieval Models

An interesting family of retrieval models can be obtained by using the KL-divergence of the query model and document model as an independent loss function in the risk minimization framework. In this case, we have,

$$\begin{aligned} r(\mathbf{d}|\mathbf{q}, \mathcal{C}, \mathcal{U}, \mathcal{S}) &\overset{\text{rank}}{\approx} D(\theta_Q || \theta_D) \\ &\overset{\text{rank}}{\approx} -\sum_{w} p(w \mid \widehat{\theta}_Q) \log p(w \mid \widehat{\theta}_D) \end{aligned}$$

Thus the ranking function is essentially the *cross entropy* of the query language model with respect to the document language model. If we let $\widehat{\theta}_Q$ be just the *empirical distribution* of the query $\mathbf{q}$, we can obtain the popular query likelihood ranking function as a special case. Thus, we see that the KL-divergence models extend the basic language modeling approach proposed in (Ponte and Croft, 1998) through the incorporation of a *query language model*, which makes it possible to treat feedback naturally as query model updating.

Feedback has so far been dealt with heuristically in the language modeling approach, usually in the form of query expansion. But such an *expansion-based* feedback strategy is not compatible with the essence of the language modeling approach – model estimation. As a result, the expanded query has to be interpreted in a *different* way than the original query

is. We proposed two *model-based* methods for performing feedback in the language modeling approach to information retrieval. One advantage of the model-based approach is that it does not cause a conceptual inconsistency when interpreting the query in the retrieval model, and it explicitly treats the feedback process as a learning process.

Specifically, let $\widehat{\theta}_Q$ be the estimated original query model and $\widehat{\theta}_{\mathcal{F}}$ be an estimated feedback query model based on feedback documents $\mathcal{F} = (d_1, d_2, ..., d_n)$, which can be the documents judged to be relevant by a user (as in the case of relevance feedback) or the top documents from an initial retrieval (as in the case of pseudo relevance feedback). Then, our new query model $\widehat{\theta}'_Q$ would be

$$\widehat{\theta}'_Q = (1 - \alpha)\widehat{\theta}_Q + \alpha\widehat{\theta}_{\mathcal{F}}$$

where, α controls the influence of the feedback model.

We propose two very different methods for estimating $\widehat{\theta}_{\mathcal{F}}$ based on feedback documents. The first method is to assume that the feedback documents are generated by a *mixture model* in which one component is the query topic model and the other is the collection language model. Given the observed feedback documents, the maximum likelihood method can be used to estimate a query topic model. The second method uses a completely different estimation criterion. It assumes that the estimated query model is the one that has the *least average KL-divergence* from the empirical word distribution of each of the feedback documents.

The two methods were evaluated with three representative large retrieval collections. Results show that both methods are effective for feedback and perform better than the Rocchio method in terms of non-interpolated average precision. These results also show that better retrieval performance can be achieved through the use of more reasonable language models and better estimation methods. In particular, we have shown that the feedback documents can be exploited to improve our estimation of the query language model(Zhai and Lafferty, 2001a).

5 Aspect Retrieval Models

While topical relevance is one of the most important factors in text retrieval, in real applications, a user often cares about other factors as well. For example, users generally would like to avoid seeing the same or similar documents again when viewing the retrieval results. So, removing redundancy in documents would be desirable. Another example is when two or more documents together can satisfy a user's information need, but when judged individually, none of them is sufficiently relevant.

To address factors like redundancy, it is necessary to break the *independent relevance* assumption. Most traditional retrieval models are based on such an independent relevance assumption, so are inadequate to address factors such as redundancy. Indeed, in these models, relevance is treated as a relationship between one *single* document and one query. The risk minimization framework allows us to "encode" user factors that we would like to

consider with the loss functions defined over a *set* of documents, so can model factors that may depend on more than one document.

We study how we can use the risk minimization framework to perform *non-traditional* ranking in the context of the *aspect retrieval* task, in which the goal is not to retrieve as many relevant documents as possible, but to retrieve as many *distinct relevant aspects* as possible.

We derive two types of aspect retrieval models, both involving a *dependent* loss function. The first type of models are to increase aspect coverage *indirectly* by reducing the redundancy among documents, and are essentially Maximal Marginal Relevance (MMR) models based on language models. The second type of models, called *Maximal Diverse Relevance* (MDR) models, are to increase aspect coverage more *directly* by modeling the hidden aspects in documents.

We propose two ways to exploit language modeling to *measure redundancy*; one is based on the KL-divergence and the other is based on mixture models. Evaluation shows that the mixture model measure is more effective in capturing redundancy, and can achieve significantly better aspect uniqueness than pure relevance ranking when re-ranking *relevant* documents. The basic idea of this mixture model measure is to represent the known information by a *known information* language model, and to assume that a new document is generated by a mixture model involving the known information model and a *background* component model. Our redundancy measure (or equivalently novelty measure) is then the estimated mixing weight on the known information model (or the background model). This redundancy measure can be useful for re-ranking any subset of documents to reduce redundancy.

For the MMR type of models, we propose four different ways to combine relevance with novelty. According to the experimental results, a direct combination of the mixture model novelty measure with the regular KL-divergence relevance score is shown to be effective in improving both aspect coverage and aspect uniqueness on the relevant data set, suggesting that it is indeed helpful for improving aspect coverage through redundancy elimination. However, this same method does not perform well on the mixed data set, and in particular, can never perform better than the baseline relevance ranking, suggesting that the redundancy elimination helps improve aspect coverage only when the relevance ranking is accurate. Overall, the MMR type of models appear to be not very effective in improving aspect performance on the mixed data. This may be because the amount of redundancy among relevant documents is relatively low compared with the amount of non-relevant documents in our data set. Since these models are designed to exploit redundancy elimination, they effectiveness is best demonstrated through a data set with a high density of redundancy. Further experiments with synthetic data would allow us to test their effectiveness more fairly.

For the MDR models, we derive a general aspect retrieval function based on *aspect KL-divergence*. The assumed generative model is illustrated in Figure 2. We assume that there exist a space of A aspects, $\boldsymbol{\tau} = (\tau_1, ..., \tau_A)$, each τ_i is characterized by a unigram language model. We further assume that a user, with an interest in retrieving documents to cover some of these A aspects, would first pick a probability distribution θ_Q over the aspects, and then

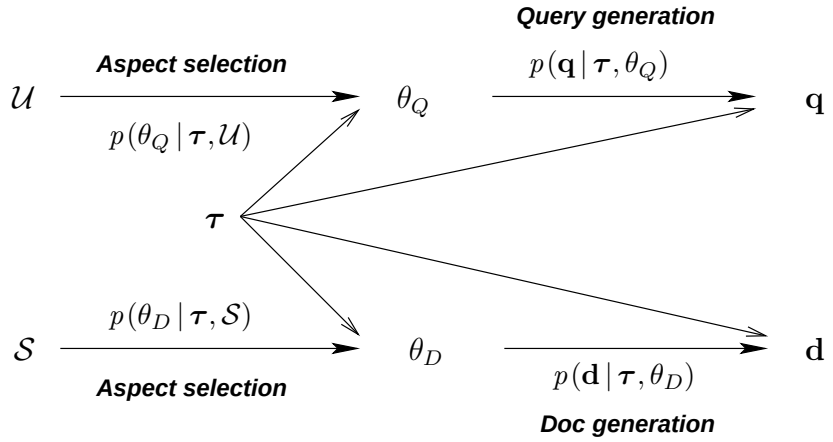

Figure 2: Aspect generative model of query $\mathbf{q}$ and document $\mathbf{d}$.

formulate a query according to a query generation model $p(\mathbf{q} \mid \boldsymbol{\tau}, \theta_Q)$. Intuitively, θ_Q encodes the user's preferences on aspect coverage, and in general, would have the probability mass concentrated on those aspects that are most interesting to the user; other non-interesting aspects may have a zero probability. Among these "interesting aspects", the distribution is generally non-uniform, reflecting the fact that some aspects are more emphasized than others. Similarly, we also assume that the author or source of a document $\mathbf{d}$ would first pick an aspect coverage distribution θ_D, and then generate $\mathbf{d}$ according to a document generation model $p(\mathbf{d} \mid \boldsymbol{\tau}, \theta_D)$.

We now consider the following loss function:

$$\begin{aligned} l(\mathbf{d}_k|\mathbf{d}_1, ..., \mathbf{d}_{k-1}, \theta, F(\mathcal{U}), F(\mathcal{S})) &= l(\mathbf{d}_k|\mathbf{d}_1, ..., \mathbf{d}_{k-1}, \boldsymbol{\tau}, \theta_Q, \theta_{D_1}, ..., \theta_{D_k}) \\ &= D(\theta_Q||\theta^{D_k}_{D_1...D_{k-1}}) \end{aligned}$$

where, $\theta^{D_k}_{D_1...D_{k-1}}$ is a weighted average of $\{\theta_{d_i}\}_{i=1}^{k}$ defined as follows:

$$p(a \mid \theta^{D_k}_{D_1...D_{k-1}}) = \frac{\mu}{k-1} \sum_{i=1}^{k-1} p(a \mid \theta_{D_i}) + (1-\mu) p(a \mid \theta_{D_k})$$

where, $\mu \in (0, 1]$ is a parameter indicating how much redundancy we would like to model.

The idea behind this loss function is the following: We expect θ_Q to give us a "measure" of which aspect is *relevant* – a high $p(a \mid \theta_Q)$ indicates that the aspect a is likely a relevant one. The loss function encodes our preferences for a *similar* "aspect coverage distribution" *collectively* given by *all* the documents $d_1, ..., d_k$. Thus, if θ_Q assigns high probabilities to some aspects, then we would expect to these (presumably relevant) aspects to be covered "more than" other aspects. The best d_k is, thus, the one that can work together with $d_1, ..., d_{k-1}$ to achieve a coverage distribution that is most similar to the desired aspect coverage of the

query, i.e., $p(a \mid \theta_Q)$. The parameter μ controls how much we would rely on the previously picked documents $d_1, ..., d_{k-1}$ to cover the aspects. If we do not rely on them (i.e., $\mu = 0$), we will be looking for a d_k that best covers all the relevant aspects by itself. On the other hand, if $\mu > 0$, part of the coverage would have been explained by the previously picked documents, and the best d_k would be the one that best covers those "under-covered" relevant aspects. Essentially, we are searching for the $\mathbf{d}_k$ that best supplements the coverage provided by the previously picked documents with respect to the desired coverage θ_Q.

We explore two variants of this aspect generative model – the *Probabilistic Latent Semantic Indexing* (PLSI)(Hofmann, 1999) and the *Latent Dirichlet Allocation* (LDA) approach(Blei et al., 2001). Evaluation shows that, for both PLSI and LDA, our loss function is effective in capturing the dependency among documents, and the optimal performance is obtained through the use of a *non-zero* novelty weight, which means that the goal of *collectively* covering the aspects with multiple documents is effectively reflected in our loss function. One interesting observation is that both PLSI and LDA have been able to improve the aspect performance even when the relevance precision decreases. This means that while these aspect models lose some relevant documents on the top, they could *selectively* keep on top those *good relevant documents* that better cover *more aspects*. We also see that, using a KL-divergence function on the *aspect space*, our relevance precision with these aspect models tends to be worse than that of applying the KL-divergence function to the *word space*, and as a result, the absolute performance is generally not as strong as the baseline relevance ranking. Nevertheless, we have shown that it is possible to for the PLSI model to perform better than the relevance ranking baseline if we further improve the basic model, e.g., by incorporating a background aspect. This is very encouraging, especially because the aspect performance is increased even when the relevance precision is decreased, which indicates the model's effectiveness in modeling the aspects. Overall, we have only touched a very basic formulation of the aspect models, especially in the case of LDA, a lot of approximation and simplification have been assumed in our experiments. Further study of these models is clearly needed to thoroughly understand these methods and their effectiveness in performing the aspect retrieval task.

6 Conclusions

In this thesis we have presented a new general probabilistic framework for text retrieval based on Bayesian decision theory. In this framework, queries and documents are modeled using statistical language models, user preferences are modeled through loss functions, and retrieval is cast as a risk minimization problem. This risk minimization framework not only unifies several existing retrieval models within one general probabilistic framework, but also facilitates the development of new principled approaches to text retrieval through the use of statistical language models. We have explored several interesting special cases of the framework.

While traditional retrieval models rely heavily on ad hoc parameter tuning to achieve satisfactory retrieval performance, the use of language models in the risk minimization frame-

work makes it possible to exploit statistical estimation methods to improve retrieval performance and set retrieval parameters automatically. As a special case of the risk minimization framework, we presented a two-stage language modeling approach that can achieve excellent retrieval performance without any ad hoc parameter tuning. The two-stage language models explicitly capture the different influences of the query and documents on the optimal setting of the smoothing parameters, allowing us to estimate the parameters independently according to different documents and queries.

Another advantage of using language models in retrieval is the possibility of improving retrieval performance through using improved language models and estimation methods. As another special case of the risk minimization framework, we derive a Kullback-Leibler divergence retrieval model that can exploit feedback documents to improve the estimation of query models. Feedback has so far been dealt with heuristically in the language modeling approach. The KL-divergence model provides a more natural way of performing feedback by treating it as query model updating. We proposed two specific query model updating algorithms based on feedback documents. Evaluation indicates that both algorithms are effective for feedback.

The risk minimization retrieval framework further allows for incorporating user factors beyond the traditional notion of topical relevance. We present language models that can capture redundancy and sub-topics in documents, and study loss functions that can rank documents in terms of both relevance and sub-topic diversity. Evaluation shows that the proposed language models can effectively capture redundancy and can outperform the relevance-based ranking method for the aspect retrieval task.

The risk minimization framework opens up many new possibilities for developing principled approaches to text retrieval, and serves as a general framework for applying statistical language models to text retrieval. The special cases explored in this thesis represent only a small step in exploring the full potential of the risk minimization framework. There are many interesting future research directions, including

Automatic parameter setting: It is always a major scientific challenge to set retrieval parameters automatically without relying on heavy experimentation. One major advantage of using language models is the possibility of estimating retrieval parameters completely automatically. The two-stage smoothing method, along with its parameter estimation methods, is a promising small step toward this goal. However, the two-stage smoothing method is a relatively simple model, which does not take advantage of feedback. It would be very interesting to study how to estimate parameters in more powerful models such as those based on feedback documents.

Document structure analysis: A common assumption in text retrieval is to treat document as the information unit. However, the relevant information often only represents a small part of a long document. Thus, it is desirable to go further than just retrieving a relevant document, and to *locate* the real relevant information in a relevant document. Traditionally, heuristic approaches have been applied to retrieve passages, which can be regarded as a step toward this goal. But the boundary of relevant information may vary according to different queries, thus it would be interesting to explore a dynamic approach where passage

segmentation is integrated with the retrieval function. With the risk minimization framework, it is possible to incorporate segmentation into the loss function and exploit language models such as Hidden Markov Models to model the document structure.

Aspect retrieval models: The aspect retrieval problem provides an interesting setting for studying redundancy elimination and sub-topic modeling in text retrieval, and is often a more accurate formulation of the retrieval problem when high recall is preferred. In this thesis, we have made some preliminary exploration of language models in the context of the aspect retrieval problem, but there are still many open issues to further explore. In particular, the Latent Dirichlet Allocation model has a great potential for modeling the hidden aspects directly, and is certainly worth further exploring. Another interesting research direction is to develop aspect retrieval models that can support aspect-based feedback.

Interactive retrieval models: In a real retrieval situation, the goal of satisfying a user's information need is generally accomplished through a series of interactions between the user and the retrieval system. Such an interactive retrieval process suggests that, at any moment, the retrieval decision may (and probably should) depend on the previous actions that the system and the user have taken. For example, the documents that a user has already seen should be considered in order to avoid redundancy. Indeed, in general, all the information about the previous actions should be utilized to provide a context for making the current retrieval decision. It would be very interesting to extend the risk minimization framework to formalize the interactive retrieval process as a sequential decision problem, and to base retrieval on optimizing the *global* and *long term* utility over a sequence of retrieval interactions.

References

Berger, J. (1985). *Statistical decision theory and Bayesian analysis*. Springer-Verlap.

Blei, D., Ng, A., and Jordan, M. (2001). Latent dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*.

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR 1998*.

Fuhr, N. (1992). Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of ACM SIGIR 1999*, pages 50–57.

Kwok, K. L. (1995). A network approach to probabilistic information retrieval. *ACM Transactions on Office Information System*, 13:324–353.

Lafferty, J. and Zhai, C. (2001a). Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR'2001*, pages 111–119.

Lafferty, J. and Zhai, C. (2001b). Probabilistic IR models based on query and document generation. In *Proceedings of the Language Modeling and IR workshop*. Extended abstract.

Maron, M. E. and Kuhns, J. L. (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, pages 216–244.

Ponte, J. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the ACM SIGIR*, pages 275–281.

Robertson, S. and Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146.

Robertson, S. E. (1977). The probability ranking principle in IR. *Journal of Documentation*, 33(4):294–304.

Robertson, S. E., van Rijsbergen, C. J., and F.Porter, M. (1981). Probabilistic models of indexing and searching. In et al., O. R. N., editor, *Information Retrieval Research*, pages 35–56. Butterworths.

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

Salton, G. and McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, (11):613–620.

Turtle, H. and Croft, W. B. (1991). Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222.

van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworths.

van Rijsbergen, C. J. (1986). A non-classical logic for information retrieval. *The Computer Journal*, 29(6):481–485.

Voorhees, E. and Harman, D., editors (2001). *Proceedings of Text REtrieval Conference (TREC1-9)*. NIST Special Publications. http://trec.nist.gov/pubs.html.

Wong, S. K. M. and Yao, Y. Y. (1995). On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems*, 13(1):69–99.

Zhai, C. and Lafferty, J. (2001a). Model-based feedback in the KL-divergence retrieval model. In *Tenth International Conference on Information and Knowledge Management (CIKM 2001)*, pages 403–410.

Zhai, C. and Lafferty, J. (2001b). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR'2001*, pages 334–342.

Zhai, C. and Lafferty, J. (2002). Two-stage language models for information retrieval. In *Proceedings of SIGIR'2002*. To appear.