Discriminative Query Models

Saar Kuzi

Discriminative Query Models

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Information
Management Engineering

Saar Kuzi

Submitted to the Senate
of the Technion — Israel Institute of Technology
Adar 5777 Haifa Februray 2017

This research was carried out under the supervision of Prof. Oren Kurland in the Faculty of Industrial Engineering and Management.

Some results in this thesis have been published as an article by the author and research collaborators in a conference during the course of the author's graduate studies period, the most up-to-date version of which being:

Saar Kuzi, Anna Shtok, and Oren Kurland. Query anchoring using discriminative query models. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, pages 219-228. ACM, 2016.

The generous financial help of the Irwin and Joan Jacobs Fellowship is gratefully acknowledged.

The generous financial help of the Technion is gratefully acknowledged.

Acknowledgements

I would like to thank my advisor Prof. Oren Kurland for his thoughtful guidance, support, and encouragement. Oren's expertise and enthusiasm for research have been a great source of inspiration for me.

The work in this thesis was done in collaboration with Dr. Anna Shtok. I would like to thank Anna for her great ideas, help, and support. It was a great pleasure to work with her.

This work was supported in part by the Israel Science Foundation (grant no. 433/12), and the Technion-Microsoft Electronic Commerce Research Center.

Finally, I would like to thank my parents Carmit and Yitzhak for their support and for always believing in me.

Contents

List of Figures

List of Tables

Abstract

Search engines are crucial tools nowadays, given the plethora of data available, as they facilitate the extraction of relevant information from large collections of digital data. The *ad hoc retrieval* task, performed by search engines, is the focus of this work. The task is to rank the documents in a collection by their relevance to an information need represented by the user's query. Modeling the presumed information need of the user is one of the challenges of the ad hoc retrieval task. To that end, different query models are often induced using the original query.

Queries in many cases do not represent the information need of the user effectively. A case in point, queries tend to be short and hence there might be a vocabulary mismatch between them and the relevant documents. Consequently, using a query model that relies solely on the user's query may result in poor retrieval performance. Various query models were devised with the goal of serving as a more effective representation of the information need.

Pseudo-feedback-based query models are induced from a result list of the documents most highly ranked by initial search performed for the query. The underlying assumption is that high ranked documents are more likely to be relevant to the query than low ranked ones. Pseudo-feedback-based query models may bridge the vocabulary mismatch by assigning high importance to terms that are presumably related to the information need.

However, since the result list often contains much non-relevant information, the induced query model may drift away from the information need [28]. Hence, various techniques, often referred to as *query anchoring*, are used for ameliorating potential query drift. For instance, interpolating the query model with a model of the original query is a common query anchoring practice.

We present a novel *unsupervised* discriminative query model that can be used, by several methods proposed herein, for query anchoring of existing query models. The model is induced from the result list using a learning-to-rank approach and constitutes a discriminative term-based representation of the initial ranking. We show that applying our methods to effective generative query models can improve retrieval performance.

Chapter 1

Introduction

With the growth of volumes of digital data in recent years, search engines have become valuable now more than ever as they facilitate the finding of relevant information in large collections of digital data. Such collections are comprised, for example, from documents, videos, or images. The focus of this work is on the *ad hoc retrieval* task, performed by search engines. The goal is to retrieve documents that are relevant to the user's information need as expressed by a query from a collection of documents [31]. One of the challenges in ad hoc retrieval is to model the presumed information need of the user. To that end, various query models were developed. These models are often induced using the original query (e.g., [34, 32]).

Queries, especially in the Web setting, are short on average [36]. Hence, they may not represent effectively the information need of the user. A case in point, there might be a vocabulary mismatch between the short query of the user and the relevant documents. That is, there might be relevant documents that do not contain some, or even all, query terms. For example, if the user's query is "car" and a relevant document uses the term "vehicle". Thus, using a query model that relies solely on the original query may result in poor retrieval performance.

Several query models were devised with the goal of serving as a more effective representation of the information need than models which are based only on the short query. Expanded query forms are an example of such query models [8]. Query expansion models often bridge the vocabulary mismatch by expanding the query with terms that are presumably related to the information need. Furthermore, using query models can also improve retrieval performance by attributing higher importance to terms that are more likely to effectively differentiate between relevant and non-relevant documents, and lower importance to terms that are less effective for this end (e.g., [23, 1, 42]).

Query models can be induced using an *initial result list* of the documents most highly ranked by some initial search performed in response to the query. For instance, some models utilize feedback from the user on documents in the result list. Such models, for example, can attribute high importance to frequent terms in the relevant documents, and low importance to frequent terms in non-relevant documents [29]. In cases where explicit feedback from the user is unavailable, pseudo feedback can be utilized instead [8]. In pseudo-feedback-based approaches the highly ranked documents in the initial result list are treated as relevant. The pseudo feedback assumption is that the higher a document is ranked, the higher its relevance likelihood. The models are then created using information induced from these documents.

Documents in the result list (a.k.a. the pseudo feedback list) could be non relevant, and relevant documents can contain non query-pertaining information [16, 18, 27]. Thus, a query model induced from these documents can drift away from the information need [28]; that is, the model can manifest aspects not related to the information need. Consequently, using the query model for retrieval can degrade performance, sometimes to the extent that using only the original query is more effective [2, 13]. Hence, several techniques, often referred to as *query anchoring*, have been proposed for mitigating the risk in relying on pseudo feedback. These techniques essentially use the original query as an anchor when utilizing pseudo feedback. For example, interpolating the query model with a model of the original query is a commonly used direct query anchoring technique (e.g., [29, 7, 1, 43, 26, 9]). Using the original query model as a prior for the pseudo-feedback-based query model is another example of direct anchoring [37, 38].

Indirect query anchoring techniques are based on various assumptions with regard to the pseudo feedback and its connection to the information need. For instance, clipping the query model by using only the terms to which it assigns the highest importance weights is common practice (e.g., [4, 40, 43, 1, 41]). The assumption is that these terms are the most likely to represent the information need as they represent the result list. Another indirect technique is attributing more importance to term occurrence in documents highly ranked in the result list than to that in low ranked documents [23, 1, 37, 33]. The premise is that the higher the document is ranked, the higher its relevance likelihood by the virtue of the way the result list was created; that is, in response to the query.

In this work we present a novel indirect query anchoring approach that can be applied to existing query models. The approach utilizes a novel *unsupervised* discriminative pseudo-feedback-based query model induced from the result list. The model serves as an accurate discriminative term-based representation of the *initial ranking* of the result list. As such, the model can be used by several methods proposed herein, for query anchoring.

As is often the case, we assume that the initial result list, which serves as the pseudo feedback list, was retrieved by using document-query surface-level similarities (e.g., a vector-space-based approach or a language-model-based approach). In order to induce the discriminative query model, we leverage the pseudo feedback assumption in a novel way. That is, we assume that for any pair of documents in the initial result list, the higher ranked one is more relevant. Consequently, we create a pairwise document preference from every pair of documents in the list. Then, these preferences are used in a pairwise learning-to-rank approach, namely SVMrank [20]. We *overfit* the learning-to-rank model to the ranking of the result list to produce an accurate as possible discriminative term-based representation of the ranking. Doing so yields an enriched discriminative query model which reflects the projection of the query on the corpus as manifested in the result list ranking.

It is important to mention that in this work we do not use learning-to-rank in its standard setting; that is, learning a ranking function using feature vectors of document-query pairs. Specifically, we use a learning-to-rank approach in order to induce a discriminative model based on pairwise pseudo preferences. In Section 3.1 we highlight the differences between using a learning-to-rank approach in our work and in a standard setting.

The novel discriminative query model is composed of two sets of terms: those that are positively and negatively correlated with the result list ranking. That is, terms that are positively correlated with the ranking tend to appear more in highly ranked than in lower ranked documents, and the reverse holds for the negatively correlated terms. The assumption is that terms that have strong correlation (positive or negative) with the initial ranking are more likely to effectively distinguish between relevant and non-relevant documents. We refer to positively and negatively correlated terms as positive anchors and negative anchors, respectively.

Using the novel discriminative query model, we devise a few methods for query anchoring of existing query models. For example, terms that are attributed high importance by a query model and that are positive anchors to a major extent should be rewarded. On the other hand, the importance of negative anchors in a query model should be diminished. We also present a method that uses both positive anchors and negative anchors. We demonstrate the merits of applying the methods on two highly effective generative query models in the language modeling framework: the relevance model [23, 1] and the mixture model [42]. Although these models employ several query anchoring approaches, applying our methods results in performance improvements.

Chapter 2

Background

2.1 The language modeling approach to information retrieval

Although our query model induction approach is not committed to a specific retrieval paradigm, it is convenient to present it in the language modeling framework given the large body of work on language-model-based query models [26].

2.1.1 Language models

Language models are distributions over sequences of words. That is, for any sequence of terms (e.g., a sentence) a language model assigns a probability of being generated by some language. In this work we focus on unigram language models and leave the treatment of more complex language models for future work. The underlying assumption of a unigram language model is that given a sequence of terms, $t_1, t_2, ..., t_n$, the probability of generating a term t_i does not depend on its context in the sequence. Formally, the probability that a sequence was generated by the language model is:

$$p(t_1, t_2, ..., t_n) = \prod_i p(t_i). \tag{2.1}$$

2.1.2 Retrieval approaches

Query likelihood A common language-model-based retrieval approach is the query likelihood model [34]. In order to rank the documents in the collection with respect to the query using this approach, the probability that a document d is relevant to the query q, $p(d|q)$, is used. Using Bayes rule we get that $p(d|q) = \frac{p(q|d)p(d)}{p(q)}$. Next, if we assume that the prior $p(d)$ is uniform, we get the rank equivalence[1]: $p(d|q) \stackrel{rank}{=} p(q|d)$. Finally, d is substituted by the language model from which it was presumably generated, M_d. The query likelihood approach, then, ranks documents according to the probability that the query was generated from their induced language models:

$$p(q|M_d) = \prod_{t \in q} p(t|M_d). \tag{2.2}$$

[1]Two functions, $f(q, d)$ and $g(q, d)$, are rank equivalent if the rankings, obtained by using each one of them for scoring documents in the collection, are equal.

Model comparison According to this approach [22], both the query and the documents are represented by language models, denoted M_q and M_d, respectively. Documents are ranked according to the similarity of their model with that of the query. When the model of the query is estimated using the maximum likelihood estimate, the model comparison approach induces a ranking that is equal to that induced by using the query likelihood approach [22]. M_q and M_d can be compared using cross entropy [22]:

$$CE\left(p(\cdot|M_q) \,\middle|\middle|\, p(\cdot|M_d)\right) \stackrel{def}{=} -\sum_t p(t|M_q) \log p(t|M_d); \tag{2.3}$$

lower values correspond to increased similarity. In this work we use the model comparison approach. Our goal is to devise an M_q that effectively represents the information need expressed by q.

2.1.3 Language model estimation

The maximum likelihood estimate (MLE) of term t with respect to the text (or text collection) x is:

$$p_{MLE}(t|x) \stackrel{def}{=} \frac{\mathrm{tf}(t \in x)}{|x|}; \tag{2.4}$$

$\mathrm{tf}(t \in x)$ is the number of occurrences of t in x; $|x| \stackrel{def}{=} \sum_{t' \in x} \mathrm{tf}(t' \in x)$ is x's length. Thus, $p_{MLE}(\cdot|x)$ is an unsmoothed unigram language model induced from x. Smoothing the MLE with a language model induced from the corpus is a common practice [43]. For instance, smoothing is useful for dealing with the problem of zero probabilities of terms that do not appear in the text but do appear in the corpus. We next review two common approaches for smoothing language models in information retrieval.

Jelinek-Mercer smoothing According to the Jelinek-Mercer technique [19], the smoothed probability of a term is a linear interpolation of the MLE induced from x with the MLE of the corpus D:

$$p_{JM}(t|x) \stackrel{def}{=} (1-\beta)p_{MLE}(t|x) + \beta p_{MLE}(t|D); \tag{2.5}$$

$\beta \in [0,1]$ is a free parameter.

Bayesian smoothing using Dirichlet prior Assuming a Dirichlet prior for the language model (which itself is assumed to be a multinomial distribution), the smoothed MLE is:

$$p_{Dir}(t|x) \stackrel{def}{=} \frac{\mathrm{tf}(t \in x) + \mu p_{MLE}(t|D)}{|x| + \mu}; \tag{2.6}$$

μ is a free parameter [43]. Jelinek-Mercer smoothing amounts to Dirichlet smoothing when the interpolation parameter β in Equation 2.5 is set to $\frac{\mu}{\mu+|x|}$. This technique takes also into consideration the length of the document. According to this approach, short documents are smoothed to a greater extent than long documents. The assumption is that long documents are

more likely to have richer vocabulary than short ones. In this work we use this approach for smoothing given its effectiveness when using short queries [43].

2.2 Pseudo-feedback-based query models

Based on the model comparison approach [22], the user's query q is represented by the model M_q; M_q is a unigram language model defined over the vocabulary. In general, we would like to induce a query model that would serve as a more effective representation of the underlying information need than a model based only on the terms in q which is often very short. An effective query model may, for example, bridge the vocabulary mismatch between the query and the relevant documents by assigning higher probabilities to terms in the vocabulary that are presumably related to the query.

Let $\mathcal{D}_{\text{init}}^{[k]}$ (henceforth $\mathcal{D}_{\text{init}}$) be a result list of the k documents most highly ranked by initial retrieval performed over the document corpus D in response to query q. We assume that the initial retrieval is based on document-query surface level similarities; e.g., a vector space model approach or a language modeling method [34, 22]. In this work, specifically, we compare a smoothed language model induced from a document with a query model estimated using MLE.

Information induced from $\mathcal{D}_{\text{init}}$, often referred to as the pseudo feedback result list, can be used to create a query model (e.g., an expanded query form). For example, the model can attribute high importance to terms frequent in documents in $\mathcal{D}_{\text{init}}$ but not in the corpus [4, 40, 7, 23, 42, 26, 8]. The underlying premise of this approach is that the highly ranked documents are more likely to be relevant to the query than the lower ranked ones.

Technically, in pseudo-feedback-based approaches two retrieval phases are performed. In the first phase, the documents are ranked according to the similarity of their induced language models with that of the query. In the second phase, a query model, estimated using the result list of the first phase, is used for ranking the documents. The user, however, is not aware of the first phase and is only exposed to the final result list.

As already noted, pseudo-feedback-based query models are often anchored to the query (e.g., via interpolation with the original query model) so as to mitigate the "risk" in relying on pseudo feedback; that is, documents in $\mathcal{D}_{\text{init}}$ can be non-relevant and relevant documents can contain much non query-pertaining information [16, 18, 27].

2.3 Generative query models

In this section we present in detail two generative query models: the relevance model [1], RM3, and the mixture model [42], MM. Both were shown to be highly effective pseudo-feedback-based approaches [26]. In Chapter 5 we demonstrate the effectiveness of our novel query anchoring approaches when applied to these.

Relevance model

The relevance model is based on the assumption that the query and documents relevant to the query are generated by a latent relevance language model [23]. Assuming that $\mathcal{D}_{\text{init}}$ was retrieved

using the query likelihood approach [34], which ranks document d as shown in Equation 2.2, the relevance model RM1 is defined as:

$$p(t|RM1) \stackrel{def}{=} \sum_{d \in \mathcal{D}_{\text{init}}} p_{Dir}(t|d)p(d|q); \tag{2.7}$$

$p(d|q) \stackrel{def}{=} \frac{p(q|d)}{\sum_{d' \in \mathcal{D}_{\text{init}}} p(q|d')}$ is the normalized query likelihood of d. RM1 is a linear mixture of the language models of documents in $\mathcal{D}_{\text{init}}$. The effect of high ranked documents on RM1 is greater than that of low ranked documents because the query likelihood values of documents serve as mixture weights. This differential effect was mentioned in Chapter 1 as an indirect query anchoring approach. Term clipping applied to RM1, which yields $\text{RM1}^{clipped}$, is an additional indirect query anchoring technique: assigning zero probability to all but the ν terms to which RM1 assigns the highest probability; ν is a free parameter; the probabilities of the ν terms are sum-normalized to produce a probability distribution. A third, direct query anchoring approach is applied by the RM3 relevance model [1]; namely, interpolating $\text{RM1}^{clipped}$ with the original query model (MLE) using a parameter λ:

$$p(t|RM3) \stackrel{def}{=} \lambda p_{MLE}(t|q) + (1-\lambda)p(t|RM1^{clipped}). \tag{2.8}$$

Mixture model

The mixture model [42] is based on the assumption that the terms in documents in $\mathcal{D}_{\text{init}}$ are generated by a mixture of two language models: a topic model, θ_T, and the corpus language model. To estimate θ_T, the log likelihood of the documents in $\mathcal{D}_{\text{init}}$,

$$\sum_{d \in \mathcal{D}_{\text{init}}} \sum_{t \in d} \text{tf}(t \in d) \log \big((1-\gamma)p(t|\theta_T) + \gamma p_{MLE}(t|D)\big),$$

is maximized using the EM algorithm; γ is a free parameter. In contrast to RM1, the relative ranking of documents in $\mathcal{D}_{\text{init}}$ does not affect the estimation of θ_T[2].

As is the case for the relevance model, θ_T is clipped, yielding $\theta_T^{clipped}$; a non-zero probability is assigned only to the ν terms to which θ_T assigns the highest probability and these probabilities are sum normalized. Direct query anchoring is performed via interpolation with the original query model, yielding the mixture model, MM:

$$p(t|MM) \stackrel{def}{=} \lambda p_{MLE}(t|q) + (1-\lambda)p(t|\theta_T^{clipped}). \tag{2.9}$$

Thus, while RM3 is based on three techniques for query anchoring: (i) differential impact of documents on the query model based on their query likelihood, (ii) term clipping, and (iii) interpolation with the original query model, the mixture model MM applies only the latter two. To use a query model θ (relevance model or mixture model) for ranking document d, the cross entropy (Equation 2.3) between the model and d's language model is used.

[2] A regularized mixture model [38] uses the original query model, $p_{MLE}(\cdot|q)$, as a Bayesian prior. This is yet another direct query anchoring technique. However, the retrieval performance is similar to that of using the original mixture model we discuss here [26].

Chapter 3

Model

3.1 A discriminative query model

Term clipping (applied by RM3 and MM) and differential impact of documents in $\mathcal{D}_{\text{init}}$ on the constructed query model (applied by RM3) are indirect query anchoring techniques. That is, the underlying assumptions are that (i) the terms most representative of $\mathcal{D}_{\text{init}}$ are likely to represent the information need; and (ii) the higher a document is ranked in $\mathcal{D}_{\text{init}}$, the higher its relevance likelihood. The latter is essentially the pseudo feedback assumption.

We leverage the pseudo feedback assumption in a different, novel way. Specifically, we directly utilize the premise that for *any* two documents d_1 and d_2 in $\mathcal{D}_{\text{init}}$, if d_1 is ranked higher than d_2, then d_1 is more likely to be relevant than d_2. Using the resultant pairwise document preferences in a pairwise learning-to-rank method [25], namely SVMrank [20], yields a discriminative query model. The model constitutes a discriminative term-based representation of $\mathcal{D}_{\text{init}}$'s *ranking*. As such, the model is used as a (indirect) query anchor for the generative query models by a few methods we present in Section 3.2.

The suggested discriminative model is based on the assumption that all pairwise preferences are equally important. We note that this assumption may not necessarily hold in reality. In Section 5.5 we drop this assumption by suggesting a version of the model in which differential weighting is assigned to pairs.

There are a few important differences between using SVMrank — or any other learning-to-rank method — in our work and in a standard learning-to-rank setting [25]. In a standard setting, the goal is to learn a ranking function using feature vectors that represent document-query pairs. Here, the goal is to create a term-based representation of a single given ranking. A feature vector is a query independent term-based representation of a document. Thus, the query model induction approach we present does not explicitly account for the query used to create $\mathcal{D}_{\text{init}}$.

The different goals in applying learning-to-rank in the standard setting and in our setting entail differences in the way the models are trained. In supervised models, different techniques are applied to avoid overfitting and to improve generalization from training data to unseen data. In contrast, we use SVMrank to produce for a given query an accurate representation of $\mathcal{D}_{\text{init}}$'s ranking. This representation is used for anchoring with respect to the given query, rather than for generalization to unseen queries. Finally, our approach is unsupervised in that it utilizes pairwise preferences that are based on pseudo feedback, while learning-to-rank methods are

usually used in supervised settings and utilize either relevance labels or implicit feedback (e.g., clickthrough information) [25].

Let $r(d)$ be the rank of document d in $\mathcal{D}_{\text{init}}$. The rank of the highest ranked document is 1. We use V to denote the vocabulary used in $\mathcal{D}_{\text{init}}$; i.e., the set of terms that appear in documents in $\mathcal{D}_{\text{init}}$. Document d ($\in \mathcal{D}_{\text{init}}$) is represented by the $|V|$ dimensional feature vector $\Phi(d)$ defined over V; the i'th component of $\Phi(d)$ is $\log p_{Dir}(t_i|d)$ where t_i is the i'th term in V and $p_{Dir}(t_i|d)$ is the probability assigned to t_i by d's Dirichlet smoothed language model. (Using $\log p_{Dir}(t_i|d)$ results in cross entropy semantics for the constraints presented in Equation 3.1.) We apply SVMrank to find a weight vector $\vec{w}$ defined over V that is the solution for:

$$\begin{aligned} &\text{m}\textit{inimize} && \frac{1}{2}\vec{w}\cdot\vec{w} + C\sum_{i,j}\xi_{i,j} \\ &\textit{subject to}: \\ &\forall i \forall j.\ r(d_i) < r(d_j): && \vec{w}\Phi(d_i) \geq \vec{w}\Phi(d_j) + 1 - \xi_{i,j} \\ &\forall i \forall j.\ r(d_i) < r(d_j): && \xi_{i,j} \geq 0 \end{aligned} \tag{3.1}$$

Equation 3.1 defines a soft margin SVM where $\xi_{i,j}$ are the slack variables and C is the regularization parameter. Higher values of C result in stricter adherence to the given pseudo-relevance-based pairwise preferences. As our goal is to fit the model as closely as possible to the ranking of $\mathcal{D}_{\text{init}}$, we will use in the experiments a very high value of C. An alternative hard-margin SVM formulation is not guaranteed to have a solution.

There are at most $\frac{1}{2}k(k-1)$ constraints of the form $\vec{w}\Phi(d_i) \geq \vec{w}\Phi(d_j) + 1 - \xi_{i,j}$ in Equation 3.1 where k is the number of documents in $\mathcal{D}_{\text{init}}$; these constraints correspond to all pairs of documents d_i and d_j in $\mathcal{D}_{\text{init}}$ where d_i is ranked *higher* than d_j (i.e., $r(d_i) < r(d_j)$). For pairs of documents with exactly the same initial retrieval score in $\mathcal{D}_{\text{init}}$ we do not use a constraint. The vector $\vec{w}$ can be thought of as a query model that contains positive and negative values that correspond to terms in V. The inner product $\vec{w}\Phi(d)$ serves for scoring documents. Given the definition of document feature vectors, we arrive to the following implication of using the pairwise constraints from Equation 3.1.

Let $\vec{w}_+$ be the vector obtained from $\vec{w}$ by setting to zero negative components. Let $\vec{w}_-$ be the vector obtained from $\vec{w}$ by setting to zero positive components, and taking the absolute value of negative components. Then, the pairwise constraint from Equation 3.1 amounts, in spirit[1], to:

$$\begin{aligned} &- CE(p(\cdot|\theta_{\vec{w}_+}) \mid\mid p_{Dir}(\cdot|d_i)) + CE(p(\cdot|\theta_{\vec{w}_-}) \mid\mid p_{Dir}(\cdot|d_i)) \geq \\ &- CE(p(\cdot|\theta_{\vec{w}_+}) \mid\mid p_{Dir}(\cdot|d_j)) + CE(p(\cdot|\theta_{\vec{w}_-}) \mid\mid p_{Dir}(\cdot|d_j)) + \\ &1 - \xi_{i,j}; \end{aligned} \tag{3.2}$$

$\theta_{\vec{x}}$ is a language model over V attained by applying L_1 normalization to $\vec{x}$. Since larger CE values correspond to decreased similarity, we get that the inequality holds to a larger extent

[1]We write "in spirit" as the weight vectors $\vec{w}_+$ and $\vec{w}_-$ that are parts of the solution to Equation 3.1 have to be normalized so as to yield valid probability distributions. Therefore, the values compared in the constraints in Equation 3.1 are not valid CE values. Yet, the observations made with respect to Equation 3.2, and consequently to Equation 3.1, still hold. We use the CE expressions to simplify the discussion.

(specifically, with lower values of $\xi_{i,j}$) when: (i) $p_{Dir}(\cdot|d_i)$ is high for terms with a (high) positive value in $\vec{w}$ and low for terms with a (low) negative value in $\vec{w}$; and, (ii) $p_{Dir}(\cdot|d_j)$ is low for terms with a (high) positive value in $\vec{w}$ and high for terms with a (high) negative value in $\vec{w}$.

As d_i is ranked above d_j, we attain the following result. Terms with a positive value in $\vec{w}$ are positively correlated with $\mathcal{D}_{\text{init}}$'s ranking — i.e., for a pair of documents in $\mathcal{D}_{\text{init}}$ they will tend to have more substantial presence in the document ranked higher. We refer to these terms as *positive anchors*. Accordingly, terms with negative values in $\vec{w}$ are negatively correlated with $\mathcal{D}_{\text{init}}$'s ranking, and are hence referred to as *negative anchors*.

It is important to highlight the difference between the discriminative query model, $(p(\cdot|\theta_{\vec{w}_+}), p(\cdot|\theta_{\vec{w}_-}))$, and generative query models, for example, those described in Section 2.3. A generative query model assigns high probability to terms that are presumably related to the underlying information need by the virtue of having substantial presence in $\mathcal{D}_{\text{init}}$. In contrast, the goal of the discriminative model is to represent the *ranking* of $\mathcal{D}_{\text{init}}$. Indeed, it attributes high positive importance to terms (positive anchors) whose presence in a document corresponds to higher ranking in $\mathcal{D}_{\text{init}}$, and high negative importance to terms (negative anchors) whose presence corresponds to lower ranking. We empirically demonstrate the difference between the two types of language models in Section 5.3. As a result, the generative query models and the discriminative model are of complementary nature. We leverage this fact in Section 3.2 by designing methods that use the discriminative query model to query anchor the generative query models.

3.2 Using the discriminative query model for query anchoring

We now present methods that use the discriminative query model, $(p(\cdot|\theta_{\vec{w}_+}), p(\cdot|\theta_{\vec{w}_-}))$, to query anchor the generative query models. Let θ be some generative query model. We assume that term clipping and explicit query anchoring (i.e., interpolation with the original query model) have not been applied to θ. For example, θ could be the RM1 relevance model, which is part of the RM3 relevance model, or the θ_T topic model which is part of the mixture model, MM. Refer back to Section 2.3 for details regarding these generative query models.

3.2.1 The AnchorPos method

The positive anchor model, $\theta_{\vec{w}_+}$, assigns non-zero probability to positive anchor terms which are positively correlated with $\mathcal{D}_{\text{init}}$'s ranking. Boosting the probabilities of these terms in a generative query model can serve for query anchoring. The AnchorPos method (named for anchoring using the positive anchor terms) integrates θ with $\theta_{\vec{w}_+}$ as follows. Let λ_1, λ_2 and λ_3 be free parameters of non-negative values used below to weigh different components of the proposed model; $\lambda_1 + \lambda_2 + \lambda_3 = 1$. We define the score $s(t)$ for term t:

$$s(t) \stackrel{def}{=} \lambda_2 p(t|\theta) + \lambda_3 p(t|\theta_{\vec{w}_+}).$$

Term clipping is applied by setting to non-zero the probability of only the ν terms t with the highest $s(t)$; this term set is denoted S; ν is a free parameter. The scores of the terms in S are sum-normalized to yield a valid language model ϑ_+ over the corpus vocabulary. That is,

$p(t|\vartheta_+) \stackrel{def}{=} 0$ for $t \notin S$; for $t \in S$: $p(t|\vartheta_+) \stackrel{def}{=} \frac{s(t)}{\sum_{t' \in S} s(t')}$. In addition, direct query anchoring is applied to yield the AnchorPos model:

$$p(t|\theta_{AnchorPos}) \stackrel{def}{=} \lambda_1 p_{MLE}(t|q) + (1-\lambda_1) p(t|\vartheta_+). \quad (3.3)$$

If θ is RM1 or θ_T, described in Section 2.3, we will refer to AnchorPos as operating on RM3 and MM, respectively. The reason is that for $\lambda_3 = 0$, Equation 3.3 amounts to RM3 and MM, respectively. In comparison to RM3 and MM which apply term clipping and direct query anchoring (RM3 applies in addition differential weighting of documents in $\mathcal{D}_{\text{init}}$), AnchorPos also applies anchoring using $\theta_{\vec{w}_+}$.

3.2.2 The ClipNeg method

The following method uses the negative anchor model, $\theta_{\vec{w}_-}$, to query anchor a generative query model θ. Let S_e be the e percent of terms assigned the highest $p(t|\theta_{\vec{w}_-})$ and which are not in the query q; e is a free parameter. These terms are the most negatively correlated with $\mathcal{D}_{\text{init}}$'s ranking. We select the ν terms to which θ assigns the highest probability and which are not in S_e. We sum-normalize the probabilities assigned to these terms by θ which yields the ϑ_- model. All other terms in the vocabulary are assigned a zero probability. Additional direct query anchoring, using a parameter λ, yields the ClipNeg model:

$$p(t|\theta_{ClipNeg}) \stackrel{def}{=} \lambda p_{MLE}(t|q) + (1-\lambda) p(t|\vartheta_-). \quad (3.4)$$

Thus, in comparison to the RM3 and MM generative models, which apply several previously proposed query anchoring techniques, the ClipNeg method also applies clipping of negative anchor terms. If θ is RM1 or θ_T we will refer to ClipNeg as operating on RM3 and MM, respectively. Specifically, for $e = 0$, i.e. when the negative anchor query model is not used, Equation 3.4 becomes RM3 and MM, respectively.

3.2.3 The AnchorClip method

To leverage both the positive anchor and negative anchor query models, we devise the AnchorClip method which boosts the probability of positive anchor terms and sets to zero the probability of negative anchor terms. As was the case for AnchorPos, we use the parameters λ_1, λ_2 and λ_3 ($\lambda_1 + \lambda_2 + \lambda_3 = 1$) and define the score of term t as: $\lambda_2 p(t|\theta) + \lambda_3 p(t|\theta_{\vec{w}_+})$. The ν terms with the highest scores, and which are not in S_e (the e percent of terms with the highest $p(t|\theta_{\vec{w}_-})$ and which are not in q) are selected to have a non-zero probability; the probability for all other terms is set to zero. Specifically, the scores of these terms are sum-normalized to yield the language model ϑ which is then interpolated with the original query model:

$$p(t|\theta_{AnchorClip}) \stackrel{def}{=} \lambda_1 p_{MLE}(t|q) + (1-\lambda_1) p(t|\vartheta). \quad (3.5)$$

For $e = 0$ and for $\lambda_3 = 0$, AnchorClip amounts to AnchorPos and ClipNeg, respectively.

All query models are used to rank the corpus by comparing them with Dirichlet smoothed document language models using the cross entropy (Equation 2.3).

To sum up, we suggested query anchoring approaches that utilize the discriminative model either by boosting the probabilities of positive anchors, or by clipping negative anchors in a given query model. However, it is still possible that using some of the positive anchors may result in query drift. According to our approaches, all positive anchors are treated equally and we leave the task of identifying potentially detrimental positive anchors for future work.

3.3 Utilizing a corpus-based language model

Several previous works have shown that explicit incorporation of a language model induced from the corpus for the estimation of pseudo-feedback-based query models can yield improvements in retrieval performance (e.g., [42, 38]). The corpus language model could be viewed as representing non-relevant documents as the majority of the documents in the corpus are non-relevant. Furthermore, frequent terms in the corpus are not associated with a single topic and therefore using these for retrieval may result in query drift [28]. Hence an effective query model is expected to be dissimilar to some extent from the corpus language model [42, 38]. We therefore set as a goal to study the merits of incorporating the corpus language model in the discriminative query model.

We represent the corpus D using the document that results from concatenating all documents in the corpus $d \in D$. (As we use unigram language models there is no importance to the order of concatenation.) Hereafter we will refer to this representation of the corpus as the *corpus document*. The definition of $\Phi(D)$ is the same as those for documents d. Specifically, the i'th component of $\Phi(D)$ is $\log p_{MLE}(t_i|D)$ where t_i is the i'th term in the vocabulary V. We set the value of D's rank, $r(D)$, to $\max_i r(d_i) + 1$. In other words, we place the corpus document after the last document in the initial result list $\mathcal{D}_{\text{init}}$ and assume that it is less relevant than any other document in the list. Given an initial result list of k documents, the following $2k$ constraints are added to SVMrank (refer to Equation 3.1 for the basic formulation):

$$\forall i: \qquad \vec{w}\Phi(d_i) \geq \vec{w}\Phi(D) + 1 - \xi_{i,D} \tag{3.6}$$

$$\forall i: \qquad \xi_{i,D} \geq 0$$

The resultant model serves as discriminative query model that utilizes the corpus (refer to Section 3.1 for the technical details):

$$(p(\cdot|\theta^D_{\vec{w}_+}), p(\cdot|\theta^D_{\vec{w}_-}));$$

we use D in the notation in order to distinct this model from the previously described $\theta_{\vec{w}_+}$ and $\theta_{\vec{w}_-}$ where the corpus document is not used in this manner.

To summarize, a discriminative model that takes into account the corpus language model promotes terms according to two criteria: (i) terms that can distinguish between higher and lower ranked documents, and (ii) terms that can distinguish the pseudo feedback set from the corpus.

We use the discriminative query model to query anchor generative query models using the methods described in Section 3.2. We denote the methods that use the corpus (D) language model as AnchorPos+D, ClipNeg+D, and AnchorClip+D.

Chapter 4

Related work

In this chapter we survey past work on mitigating the risk in using pseudo feedback for the estimation of a query model. Several approaches were devised in past works to this end, including: explicit query anchoring techniques, methods for improving the quality of an initial result list, methods for improving an initial pseudo-feedback-based query model, approaches for combining several query models, and methods for automatic selection between the original query and the pseudo-feedback-based query model on a per query basis.

Explicit query anchoring techniques A few approaches have been proposed previously as explicit means to keep the pseudo-feedback-based query models faithful to the original query (a.k.a. query anchoring). One commonly used approach is to clip the pseudo-feedback-based query model by using only a few presumably important terms (e.g., [4, 40, 43, 1, 41]). The assumption is that these terms are the most effective in representing the information need of the user as they represent the initial result list. Another commonly used method is query anchoring via interpolation of the pseudo-feedback-based query model with a model of the original query (e.g., [29, 7, 1, 43, 26]). The idea is to "emphasize" the terms of the original query in the resulting query model in order to ameliorate potential query drift. Attributing more importance to highly ranked documents in the initial result list (i.e., the pseudo feedback list) than to lower ranked ones when constructing the query model is an additional query anchoring approach [23, 38, 41]. The premise is that highly ranked documents are more likely to be relevant to the query. We show that using our methods in addition to these three common and effective approaches — interpolation with a model of the original query, term clipping, and differential weighting to documents — helps to further improve retrieval effectiveness.

A fusion approach Fusing the initial ranking with the ranking produced by using the pseudo-feedback-based query model was suggested for indirect query anchoring [44]. That is, a score of a document in the final result list is the fusion of its scores in the two rankings. In other words, this fusion approach rewards documents that are highly ranked in two result lists, obtained by using different representations of the presumed information need. In contrast, our methods operate on the query model by integrating it, at the model level, with a representation of the initial ranking (i.e., the proposed discriminative query model). In Section 5.2 we show that our methods substantially outperform this fusion approach [44].

Improving the quality of the initial result list There are various methods for improving the quality of the result list used for inducing the pseudo-feedback-based query model (e.g., [28, 30, 24, 17, 21]). Classification of documents in the initial result list was proposed [17]. The classifier was used in order to select documents for the pseudo feedback set that constitute a more effective input for the query model estimation. In a similar vein, the weights, attributed to documents used for pseudo-feedback-based query model construction were learned using regression [21]. The goal was to assign scores to documents that coincide with the documents' contribution to the effectiveness of the induced query model. Both works [17, 21] used features that reflect the general quality of the document, the relation between the document and the query, and the similarity of the document with other documents in the pseudo feedback set. A method for re-ranking the documents in the initial result list was proposed in [28]. Using this approach, only the highly ranked documents in the re-ranked list were used for the query model induction. Specifically, the different aspects of the query were represented by a boolean constraint in a conjunctive normal form. Documents were then ranked by the extent to which they satisfy the constraint. The premise is that a document that covers many aspects of the query is expected to be effective for the estimation of a query model. Methods for cluster-based re-sampling of documents from the initial result list were also devised [24, 30]. In one approach [24] documents in the initial result list were clustered based on lexical similarity using a soft clustering approach; that is, a document can be associated with several clusters. Then, the query model was induced using documents contained in highly ranked clusters. The underlying assumption is that the different clusters represent the different aspects of the query. Hence, documents associated with several highly ranked clusters are likely to be relevant to the information need by the virtue of covering several important aspects of the query. In another work [30], a hard clustering method was used; that is, a document can be associated with only one cluster. Specifically, the documents were clustered based on the original query terms that they contain regardless of the frequency of the query terms in the documents. The goal of [30] was to re-sample documents using the clusters in order to create a diverse sample that would faithfully represent the different aspects of the query. Our methods are complementary to these past approaches for two reasons: (i) the discriminative query model used by our methods can be induced from any ranked list, and (ii) our methods can be applied to any query model regardless of the initial ranking from which the model was induced.

Improving an initial pseudo-feedback-based query model Methods that improve an initial pseudo-feedback-based query model were also developed in previous works (e.g., [7, 5, 11]). A method for re-weighting of terms in a query model was suggested [7]; specifically, the method scored each term by its contribution to the Kullback-Leibler divergence between a pseudo-feedback-based language model and a language model representing the corpus. The assumption was that terms that can help to differentiate the vocabulary of the pseudo feedback set from that of the corpus are effective for query expansion. Motivated by this work, we show in Section 5.4 that utilizing a corpus-based language model in the estimation of the discriminative query model can further improve performance. An optimization framework for query models that takes into account not only the benefit in using a query model, but also the risk involved, was proposed [11]. Using the framework, terms in a given query model were re-weighted with the goal of

reducing the extent to which the model fails, while not hurting the average performance of the model. Another work [5] has focused on measuring the effectiveness of individual expansion terms. Specifically, a supervised term classification approach to selecting query expansion terms among those attributed the highest importance by some query model was proposed. The approach was applied on a unigram query model, but the most effective features rely on term-proximity information. In contrast, our approach is unsupervised and applies a learning-to-rank method on documents in the initial result list. Furthermore, we focus on unigram models and leave the treatment of query models that utilize term proximity information for future work. In a related vein, some work [39] showed that if the terms attributed the highest importance by a query model are clustered into two clusters, then using only the terms in one of the clusters will be much more effective for retrieval than using all terms, or using the terms in the other cluster. However, a method of selecting the cluster from the two given ones was not proposed. More generally, as in the case of methods that improve the quality of the pseudo feedback set, the discriminative query model can be applied on top of methods that improve an initial query model.

Combining several query models There have been works on integrating several query models [12, 35]. One work [12] modeled the risk in using a pseudo-feedback-based query model. Specifically, the posterior distribution over query models was estimated using re-sampling of documents from the initial result list (giving priority to highly ranked documents). Then, the mean (or mode) of the estimated distribution was used as a query model. In order to construct a more robust model, several query models (generated using variants of the original query) were combined. In another work [35] a probabilistic model that accounts for the uncertainty of the information need underlying the query was devised. Specifically, a method was developed in which multiple query models, induced using samples of documents from the initial result list, were integrated using fusion of their rankings. Several functions that assign a score to each model in the fusion were evaluated. For instance, a function that promotes query models which produce a ranking that is close to the ranking produced by the original query was shown to be effective. Both of these approaches [12, 35] are complementary to our approach as the discriminative model can also be applied on these query models.

Methods for selective query expansion on a per query basis For some queries, using a pseudo-feedback-based query model for retrieval results in performance worse than that of using only a model of the original query. Hence, methods that select one of the two for retrieval, on a per-query basis, were proposed [2, 13]. A language modeling approach for selective query expansion was developed [13]. Specifically, the similarity between a language model induced from the initial ranking and a query model induced from the pseudo-feedback-based ranking was measured. Then, this similarity score was used in order to decide whether to use the pseudo-feedback-based query model or not. The idea is that this score should indicate the extent of which the expanded query has strayed from the original one. In another work [2] the decision of whether to expand the original query was made based on a scoring function for queries that takes into account several factors such as the query length and the ratio between the frequency of query terms in the pseudo feedback set and in the corpus. In both works [2, 13], however,

the results were not conclusive. In a related vein, it was demonstrated empirically that setting the values of the free parameters of the feedback method used to construct the expanded query form on a per query basis, specifically, the number of documents and expansion terms, can yield improvements over using fixed values [3]. Another work [27], motivated by this finding, tuned the query anchoring interpolation parameter (e.g., the parameter λ used in RM3 and MM; refer to Section 2.3 for more details) on a per-query basis. However, the method was applied to true, rather than pseudo, relevant documents. We show that applying our methods on top of interpolation-based query anchoring is of merit.

Formal analysis of pseudo-feedback-based query models Formal analysis of methods for inducing pseudo-feedback-based query models, and the properties of terms that should be assigned high importance by query models, was presented [26, 10]. Our findings provide an additional, novel characterization: the importance weight of terms whose presence in documents is positively correlated with the *initial ranking* should be increased while that of terms whose presence is negatively correlated should be decreased.

Chapter 5

Evaluation

We present an evaluation of the methods from Section 3.2 that use the discriminative query model. The methods are applied to the relevance model, RM3 [1], and to the mixture model [42], MM, described in Section 2.3. These two generative query models were the most effective in a study of unigram language-model-based query models [26].

5.1 Experimental setup

The datasets specified in Table 5.1 were used for experiments. TREC123 and ROBUST are (mainly) newswire document collections, and WT10G is a Web collection. Titles of TREC topics serve for queries. Krovetz stemming and stopword removal (using the INQUERY list) were applied to documents and queries. We used for experiments the Indri toolkit (www.lemurproject.org).

The initial result list $\mathcal{D}_{\text{init}}$, which serves for pseudo feedback, is retrieved using a standard language model method [22] which uses cross entropy (see Equation 2.3): document d is scored by $-CE\left(p_{MLE}(\cdot|q) \,\middle|\middle|\, p_{Dir}(\cdot|d)\right)$. The ranking is equivalent to that produced by the query likelihood model [34] used in the relevance model. Here and after, the Dirichlet smoothing parameter, μ, is set to 1000 [43].

We use Mean Average Precision (MAP@1000) and the precision of the top-5 documents (p@5) for evaluation measures. Statistically significant differences of performance are determined using the two-tailed paired t-test at a 95% confidence level. We also report the reliability of improvement (RI) [30] for the query models: $100 \cdot \frac{|Q_+|-|Q_-|}{|Q|}$; Q is the set of queries; Q_+ and Q_- are the sets of queries for which the average precision (AP) is higher and lower, respectively, than that of the initial ranking. The RI measure quantifies the performance robustness of using a pseudo-feedback-based query model with respect to using only the query. In Section 5.2.3 we extend the robustness analysis by using risk-reward graphs [11].

As mentioned in Section 3.1, we use SVMrank [20] to construct the discriminative query model; the regularization parameter C is set to $100,000$. All other parameters of SVMrank are set to default values[1]. The resultant model is nearly a hard margin SVM fitted to $\mathcal{D}_{\text{init}}$'s ranking. Recall that our goal is to create an accurate representation of this ranking. Indeed, lower values of C resulted in less effective anchoring models. Actual numbers are omitted as they convey no additional insight.

[1] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

Collection	TREC disks	# of Docs	Topics
TREC123	Disks 1&2	741,856	51-200
ROBUST	Disks 4,5-{CR}	528,155	301-450, 601-700
WT10G	WT10g	1,692,096	451-550

Table 5.1: TREC datasets used for experiments.

Our methods apply the discriminative query model, which represents $\mathcal{D}_{\text{init}}$'s ranking, to query anchor the generative query models. Thus, they could be viewed as fusing a representation of the initial ranking with the generative query model at the language model level. Hence, we use a reference comparison, **Fusion**, that fuses the initial ranking with the generative query models at the retrieval score level [44]. Specifically, the top-1000 documents in the initial ranking are fused, using CombMNZ [15], with the top-1000 documents in a ranking produced by using the generative query model [44][2]. The idea is to reward documents highly similar both to the pseudo-feedback-based query model and to the query. The implementation details are as in [44].

Free-parameter values All the methods we consider: the generative query models RM3 and MM, our methods (AnchorPos, ClipNeg and AnchorClip) and the reference comparison Fusion incorporate free parameters. We set the values of all free parameters of each method using leave-one-out (LOO) cross validation performed over the queries in a dataset[3]. That is, the free parameters of a method for a query are set to values that optimize average performance over all other queries in the dataset. To avoid metric divergence issues, following previous recommendations in work on query expansion [14] we use the same evaluation metric (MAP or p@5) to train free-parameter values and to report the resultant performance. The following free-parameter value ranges were used. The number of documents, k, in the initial list $\mathcal{D}_{\text{init}}$ is in $\{25, 50, 100\}$. The number of terms used in the query models, ν, is in $\{25, 50, 75\}$. The mixture model parameter, γ, is in $\{0.1, 0.5, 0.9\}$. The percentage of negative anchor terms, e, clipped in ClipNeg and AnchorClip is in $\{0, 5, 10, 25, 50, 75, 100\}$ for ClipNeg and in $\{75, 100\}$ for AnchorClip[4]. The parameters λ, λ_1, λ_2 and λ_3 are set to values in $\{0, 0.2, \ldots, 1\}$.

5.2 Experimental results

5.2.1 The discriminative model

We first study the extent to which the discriminative query model represents the initial ranking of the list, $\mathcal{D}_{\text{init}}$, from which it is constructed. To that end, we re-rank $\mathcal{D}_{\text{init}}$ using the model. The score of d ($\in \mathcal{D}_{\text{init}}$) is the interpolation of d's similarity with the positive anchor model and

[2]CombMNZ was more effective in our setting than the alternative interpolation-based fusion method [44].

[3]The performance of *all* methods when using 10-fold cross validation was sometimes slightly lower than that of using LOO; but, most differences were statistically indistinguishable, and the relative performance patterns were the same.

[4]Since $e \neq 0$ for AnchorClip, we enforce clipping. We show in Section 5.2.3 that clipping a low percentage of negative anchor terms yields worse performance than clipping a high percentage.

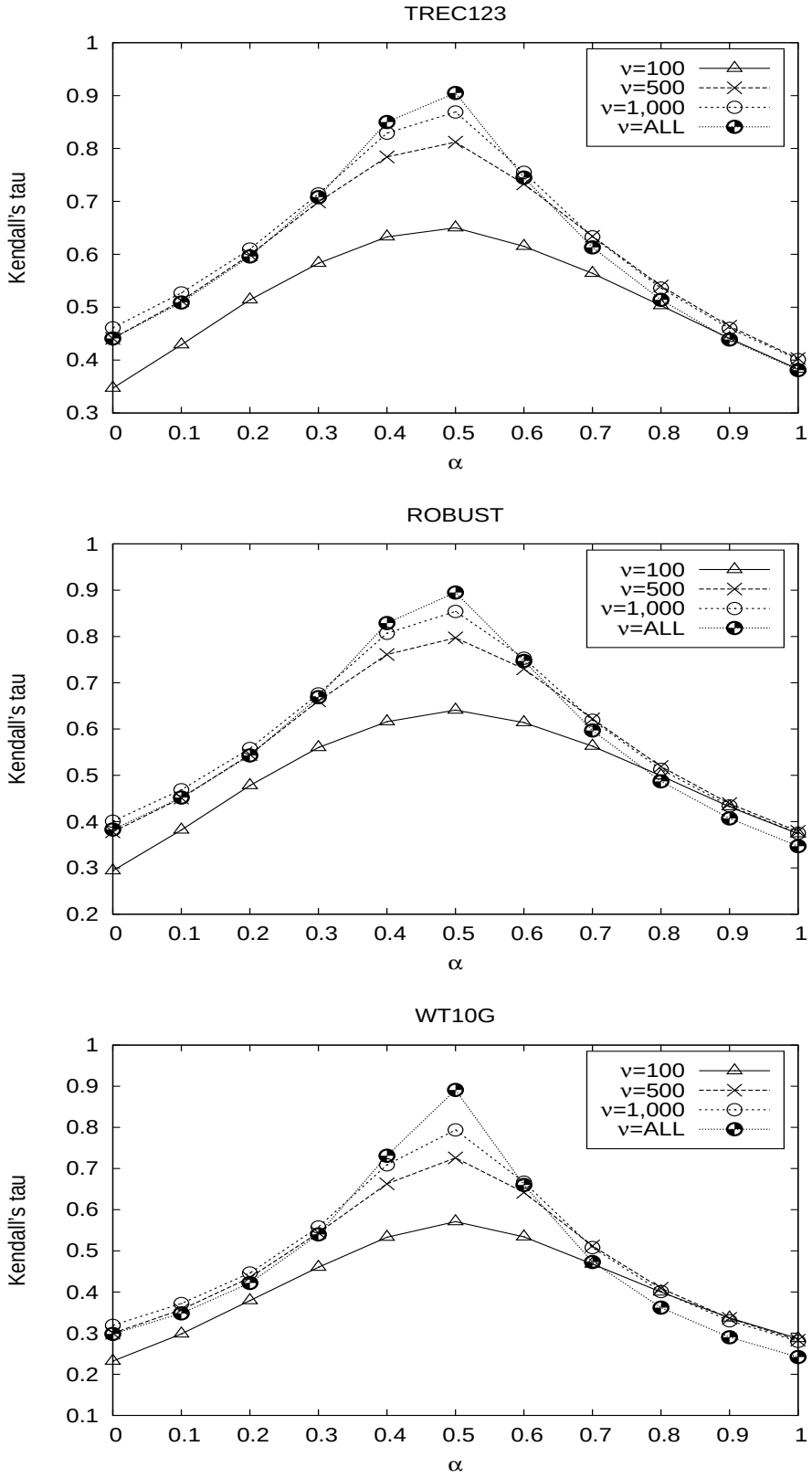

Figure 5.1: Using the discriminative query model to re-rank an initial list of 100 documents from which it is induced. The Kendall-τ between the initial ranking (init) and the re-ranking is reported. The positive and negative anchor models are clipped to use ν terms; ν =ALL means no clipping. Note: figures are not to the same scale.

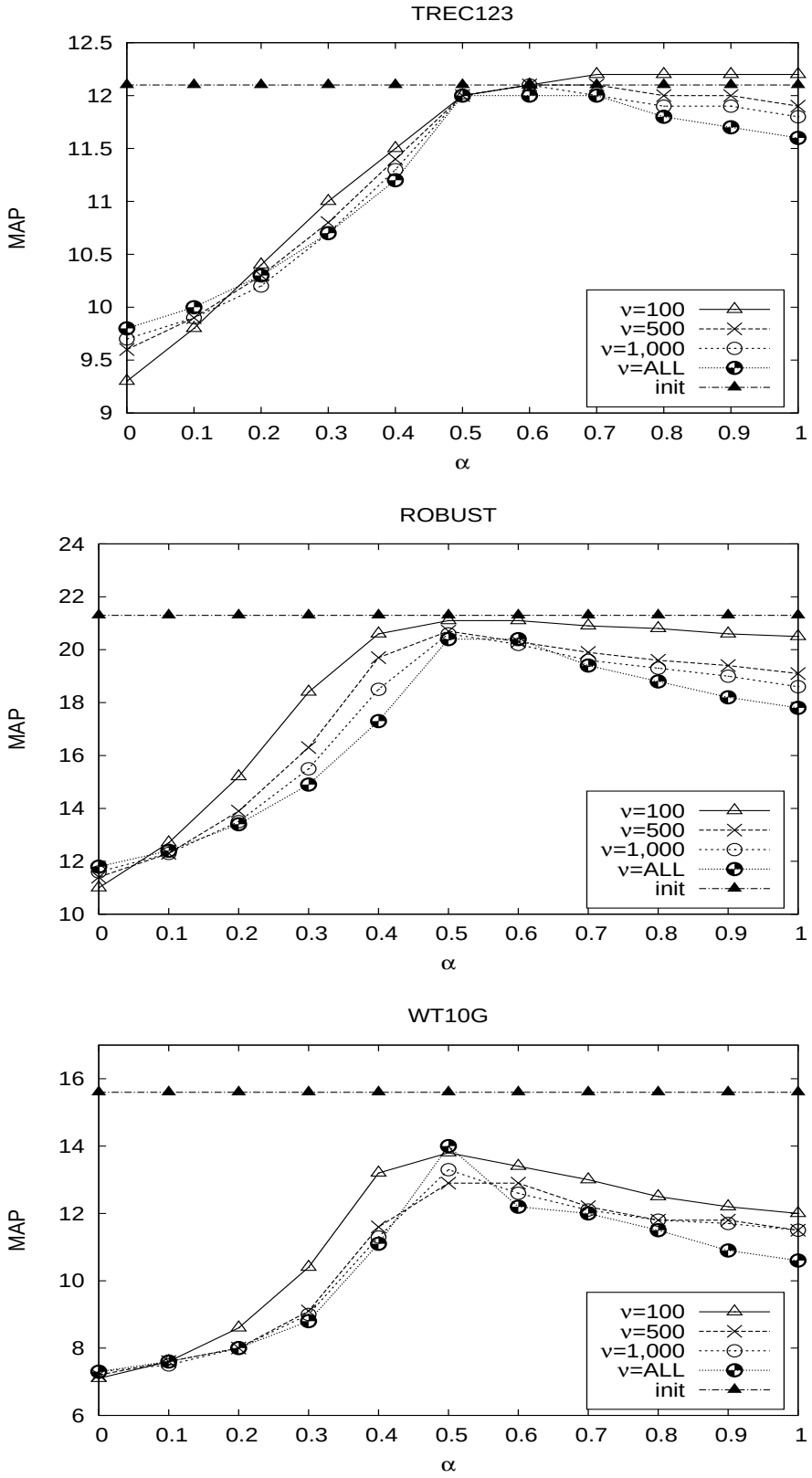

Figure 5.2: Using the discriminative query model to re-rank an initial list of 100 documents from which it is induced. MAP(@100) of the initial ranking (init) and the re-ranking is reported. The positive and negative anchor models are clipped to use ν terms; ν =ALL means no clipping. Note: figures are not to the same scale.

dissimilarity with the negative anchor model:

$$-\alpha CE(p(\cdot|\theta_{\vec{w}_+}) \;||\; p_{Dir}(\cdot|d)) + (1-\alpha)CE(p(\cdot|\theta_{\vec{w}_-}) \;||\; p_{Dir}(\cdot|d));$$

α ($\in \{0, 0.1, \ldots, 1\}$) is a free parameter; $\theta_{\vec{w}_+}$ and $\theta_{\vec{w}_-}$ are clipped to use the ν terms to which they assign the highest probabilities. We report the Kendall-τ between the re-ranking of $\mathcal{D}_{\text{init}}$ and its initial ranking as a function of α and ν. The correlation takes values in $[-1, 1]$ where -1 and 1 represent perfect negative and positive correlation, respectively. The analysis, presented in Figure 5.1, is applied to $\mathcal{D}_{\text{init}}$ of $k = 100$ documents. In Figure 5.2 we also report MAP(@100) for the rankings[5].

We see in Figure 5.1 that the highest correlation is always attained for $\alpha = 0.5$; increasing the number of terms, ν, results in higher correlation. Specifically, using all terms in documents in $\mathcal{D}_{\text{init}}$ with $\alpha = 0.5$ yields a correlation of around 0.9 which is very high. Thus, we see that by attributing the same importance to the positive and negative anchor models (i.e., $\alpha = 0.5$) the discriminative model ($\theta_{\vec{w}_+}$,$\theta_{\vec{w}_-}$) becomes quite an accurate representation of $\mathcal{D}_{\text{init}}$'s ranking.

We also see in Figure 5.2 that for $\alpha = 0.5$, and regardless of the number of terms used, re-ranking performance can be quite close, or identical, to that of the initial ranking. This finding resonates with the high correlation between the rankings. Furthermore, low values of α (< 0.5) are more detrimental for performance than high values (> 0.5). This implies that using the positive anchor model is somewhat more effective in improving performance than the negative anchor model. We further support this finding below.

We see in Figure 5.2 that for $\alpha \neq 0.5$ a low number of terms often yields better performance than a high number. This finding implies that the terms assigned with the highest probability by the positive and negative anchor models are the most positively and negatively correlated, respectively, with the initial ranking. Indeed, terms with a high absolute value in $\vec{w}$ in SVMrank are the most influential in establishing the decision boundary.

5.2.2 Main result

Table 5.2 presents the performance comparison of our ClipNeg and AnchorPos methods with the initial ranking, the generative model on which they are applied (RM3 and MM) and the Fusion reference comparison. The performance of the AnchorClip method, which integrates ClipNeg and AnchorPos, is studied below. We see in Table 5.2 that all methods outperform in most cases — often to a statistically significant degree — the initial ranking.

Our AnchorPos and ClipNeg methods improve over the generative query model on which they are applied (RM3 or MM) in terms of MAP and p@5 in most relevant comparisons (3 corpora $\times$ 2 evaluation measures $\times$ 2 generative models); the majority of MAP improvements for AnchorPos are statistically significant. Furthermore, the RI of ClipNeg and AnchorPos is in most cases higher than that of the generative model. These findings attest to the merits of using our discriminative query model to query anchor generative query models which already apply a few query anchoring techniques. We also see in Table 5.2 that using positive anchor terms (AnchorPos) is almost always more effective in terms of retrieval effectiveness (MAP and

[5]This is the only case where MAP@100 rather than MAP@1000 is used, since we focus here on re-ranking a list of 100 documents.

	TREC123			ROBUST			WT10G		
	MAP	p@5	RI	MAP	p@5	RI	MAP	p@5	RI
init	23.0	52.7	–	24.9	47.6	–	19.8	35.5	–
	Relevance Model								
RM3	28.4^{i}	$\mathbf{57.7}^{i}$	48.0	28.2^{i}	48.6	30.9	$\mathbf{21.9}^{i}$	37.7	5.2
Fusion	$27.0^{i,g}$	54.1^{g}	46.7	27.7^{i}	48.9^{i}	**39.0**	$20.8^{i,g}$	37.9	7.2
ClipNeg	$28.8^{i,g}_{f}$	$\mathbf{57.7}^{i}_{f}$	52.0	$28.9^{i,g}_{f}$	$50.0^{i,g}$	36.5	21.6^{i}	**38.1**	**8.2**
AnchorPos	$\mathbf{29.2}^{i,g}_{f,c}$	$\mathbf{57.7}^{i}_{f}$	**53.3**	$\mathbf{29.6}^{i,g}_{f}$	$\mathbf{50.1}^{i,g}$	31.3	21.8^{i}	37.7	−2.1
	Mixture Model								
MM	28.1^{i}	55.1	38.7	27.6^{i}	48.8	25.3	20.8^{i}	**36.3**	12.4
Fusion	$27.7^{i,g}$	53.6^{g}	39.3	27.8^{i}	48.4	29.7	20.2^{g}	35.1	−3.1
ClipNeg	28.3^{i}	53.3	38.7	27.6^{i}	46.1^{g}_{f}	18.9	21.0^{i}_{f}	33.2^{g}	**16.5**
AnchorPos	$\mathbf{29.1}^{i,g}_{f,c}$	**55.9**	**40.0**	$\mathbf{29.2}^{i,g}_{f,c}$	$\mathbf{50.0}^{i}_{c}$	**30.9**	$\mathbf{21.3}^{i}_{f}$	34.0	9.3

Table 5.2: Main result. Boldface: best result in a column in a generative model block. Statistically significant differences with the initial ranking (init), generative model (RM3 or MM), Fusion and ClipNeg are marked with 'i', 'g', 'f' and 'c', respectively.

p@5) than using negative anchor terms (ClipNeg). More generally, in most cases, AnchorPos is the most effective method in Table 5.2 in terms of MAP and p@5. In terms of RI, neither AnchorPos nor ClipNeg dominates the other.

The ClipNeg and AnchorPos methods outperform the Fusion reference comparison, in terms of MAP and p@5, in a vast majority of the cases; many of the improvements posted by AnchorPos are statistically significant. In most cases, Fusion is outperformed (MAP and p@5) by the generative query model on which it is applied (RM3 and MM), while it often improves RI. Indeed, the goal of Fusion is to improve performance robustness even at the expense of hurting average retrieval effectiveness [44]. Yet, the RI of Fusion is in most cases inferior to that of AnchorPos.

Table 5.2 shows that the effectiveness of the initial ranking, which is used to induce the generative models and our discriminative model that is applied to the generative models, is much lower for WT10G than for TREC123 and ROBUST. Consequently, the improvements posted by the generative models over the initial ranking, and those posted by our ClipNeg and AnchorPos methods over the generative models, are much smaller for WT10G than those for TREC123 and ROBUST. In some cases for WT10G, ClipNeg and AnchorPos are outperformed by the generative model although the difference is statistically significant only in a single case (for ClipNeg). Still, ClipNeg and AnchorPos are more effective in terms of MAP, and to a statistically significant degree, than the initial ranking for WT10G. For reference comparison, the Fusion method is almost always outperformed by the generative models for WT10G — statistically significantly so in two cases. Another related finding about the differences between WT10G and the other two corpora is that the RI values of all pseudo-feedback-based methods are much smaller for WT10G. Indeed, we found that the (learned) value of the query-anchoring parameter (i.e., the weight of the original query model) in all pseudo-feedback-based models is consistently higher for WT10G than for the other two corpora. This further attests to the overall limited effectiveness of using pseudo feedback for WT10G.

AnchorClip

The AnchorClip method, presented in Section 3.2, integrates the ClipNeg and AnchorPos methods by boosting the probabilities of positive anchor terms as in AnchorPos and clipping negative

	TREC123			ROBUST			WT10G		
	MAP	p@5	RI	MAP	p@5	RI	MAP	p@5	RI
Relevance Model									
ClipNeg	28.8	**57.7**	52.0	28.9	50.0	**36.5**	21.6	**38.1**	**8.2**
AnchorPos	29.2^c	**57.7**	**53.3**	**29.6**	50.1	31.3	**21.8**	37.7	−2.1
AnchorClip	**29.4**^{c}	56.7	52.0	29.5	**51.1**	34.1	21.1	37.3	2.1
Mixture Model									
ClipNeg	28.3	53.3	38.7	27.6	46.1	18.9	21.0	33.2	**16.5**
AnchorPos	29.1^c	55.9	**40.0**	**29.2**^{c}	50.0^c	**30.9**	21.3	**34.0**	9.3
AnchorClip	**29.4**^{c}	**56.4**^{c}	38.7	29.1^c	**50.4**^{c}	28.1	**21.5**	32.6	8.2

Table 5.3: Comparison of AnchorClip with ClipNeg and AnchorPos. Boldface: best result in a column in a generative model block. Statistically significant differences with ClipNeg are marked with 'c'. There are no statistically significant differences between AnchorClip and AnchorPos.

anchor terms as in ClipNeg. Table 5.3 presents a performance comparison of AnchorClip with ClipNeg and AnchorPos.

We see that AnchorClip outperforms (in terms of MAP and p@5) ClipNeg in most cases; several of the improvements are statistically significant. However, neither of AnchorClip and AnchorPos dominates the other; specifically, the performance differences between these methods are never statistically significant. This finding implies that that there is no clear merit in clipping negative anchor terms in addition to boosting the probabilities of positive anchor terms. We hasten to point out, however, that this finding could potentially be attributed to the fact that AnchorClip incorporates more free parameters than AnchorPos (specifically, the percentage of negative anchor terms to clip). Setting the values of all these parameters using cross validation with the relatively small query sets at hand can fall short. Indeed, experiments — actual numbers omitted as they convey no additional insight — show that if free-parameter values are set to optimize average performance over all queries in a dataset, then AnchorClip yields consistent improvements over AnchorPos, albeit not statistically significant.

5.2.3 Further analysis

We next turn to further explore the utilization of positive anchor terms (the AnchorPos method) and negative anchor terms (the ClipNeg method).

AnchorPos

Setting $\lambda_2 = 0$ in AnchorPos yields a query model, **Q+Pos**, that interpolates the original query model with the clipped positive anchor model; the generative query model is not used. Setting $\lambda_1 = 0$ results in the **M+Pos** method which integrates the positive anchor model with the generative query model; term clipping is applied but not interpolation with the original query model. Table 5.4 presents the performance of these specific cases of AnchorPos.

In most cases, Q+Pos statistically significantly outperforms (MAP and p@5) the initial ranking and is outperformed (often, statistically significantly so) by the generative query model (RM3 and MM). The latter finding comes as no surprise as the goal of the discriminative query model is to accurately represent the ranking of the initial result list rather than the information need. Yet, the superiority of Q+Pos to the initial ranking provides further support to the merits of using positive anchor terms for retrieval.

	TREC123			ROBUST			WT10G		
	MAP	p@5	RI	MAP	p@5	RI	MAP	p@5	RI
init	23.0	52.7	–	24.9	47.6	–	19.8	35.5	–
Relevance Model									
RM3	28.4^{i}	$\mathbf{57.7}^{i}$	48.0	28.2^{i}	48.6	30.9	$\mathbf{21.9}^{\mathbf{i}}$	**37.7**	5.2
AnchorPos	$\mathbf{29.2}^{i,g}$	$\mathbf{57.7}^{i}$	**53.3**	$\mathbf{29.6}^{i,g}$	$\mathbf{50.1}^{i,g}$	**31.3**	21.8^{i}	**37.7**	−2.1
Q+Pos	$25.3^{i,g}_{a}$	55.3^{i}	28.7	$26.6^{i,g}_{a}$	49.7^{i}	4.8	20.4^{g}_{a}	36.3	**10.3**
M+Pos	$27.4^{i}_{a,p}$	56.8^{i}	17.3	27.0^{i}_{a}	$43.9^{i,g}_{a,p}$	3.6	$15.5^{i,g}_{a,p}$	34.6	−50.5
Mixture Model									
MM	28.1^{i}	55.1	38.7	27.6^{i}	48.8	25.3	20.8^{i}	**36.3**	**12.4**
AnchorPos	$\mathbf{29.1}^{i,g}$	**55.9**	**40.0**	$\mathbf{29.2}^{i,g}$	$\mathbf{50.0}^{i}$	**30.9**	$\mathbf{21.3}^{i}$	34.0	9.3
Q+Pos	$25.3^{i,g}_{a}$	55.3^{i}	28.7	26.6^{i}_{a}	49.7^{i}	4.8	20.4	**36.3**	10.3
M+Pos	$26.8^{i,g}_{a}$	$48.4^{i,g}_{a,p}$	2.7	25.9^{g}_{a}	$45.9^{g}_{a,p}$	−1.2	$14.7^{i,g}_{a,p}$	$30.3^{i,g}_{a}$	−45.4

Table 5.4: AnchorPos and its two specific cases: Q+Pos and M+Pos. The performance of Q+Pos is identical for RM3 and MM as it does not incorporate the generative query model. The best result in a column in a generative model block is boldfaced. Statistically significant differences with the initial ranking, the generative query model (RM3 or MM), AnchorPos and Q+Pos are marked with 'i', 'g', 'a' and 'p', respectively.

The MAP performance of M+Pos is often in-between that of the initial ranking and the generative query model except for WT10G; yet, the p@5 performance of M+Pos is almost always below that of the initial ranking. These findings show that direct anchoring using the original query, which is not applied in M+Pos, is highly important. More generally, the superiority of AnchorPos to Q+Pos and M+Pos attests to the merits of applying both the discriminative model and direct query anchoring to a generative model.

To further study the performance robustness of AnchorPos, in Figure 5.3 we present its MAP risk-reward curves [11] when applied to RM3 and MM and those of the generative models themselves. A curve is created by varying the value of the query anchoring parameter (λ for RM3 and MM, and λ_1 for AnchorPos) from 1 (using only the original query) to 0 (no direct query anchoring) with .2 decrement[6]; all other free parameters are set here to optimize MAP over all queries so as to study the potential risk-reward tradeoffs of the models. The x-axis (R-loss) is the resultant difference, over all queries for which the difference is non negative, between the number of relevant documents retrieved using only the original query and using the pseudo-feedback-based query model; the y-axis (reward) is the percentage of MAP improvement over all queries of applying the query model with respect to using only the original query.

Figure 5.3 shows that in most cases the curves for AnchorPos dominate those for the generative models; i.e., for the same value of query anchoring parameter, the point on the curve of AnchorPos would be to the left of, and higher than, that for the generative model. In the other cases, AnchorPos posts higher reward at the expense of higher risk for the same value of query anchoring parameter. These findings further support the merits of using positive anchor terms.

[6]The only case where the loss did not increase with decreasing the value of the anchoring parameter was for applying AnchorPos to MM in WT10G: the second and third points on the curve correspond to 0.6 and 0.8, respectively.

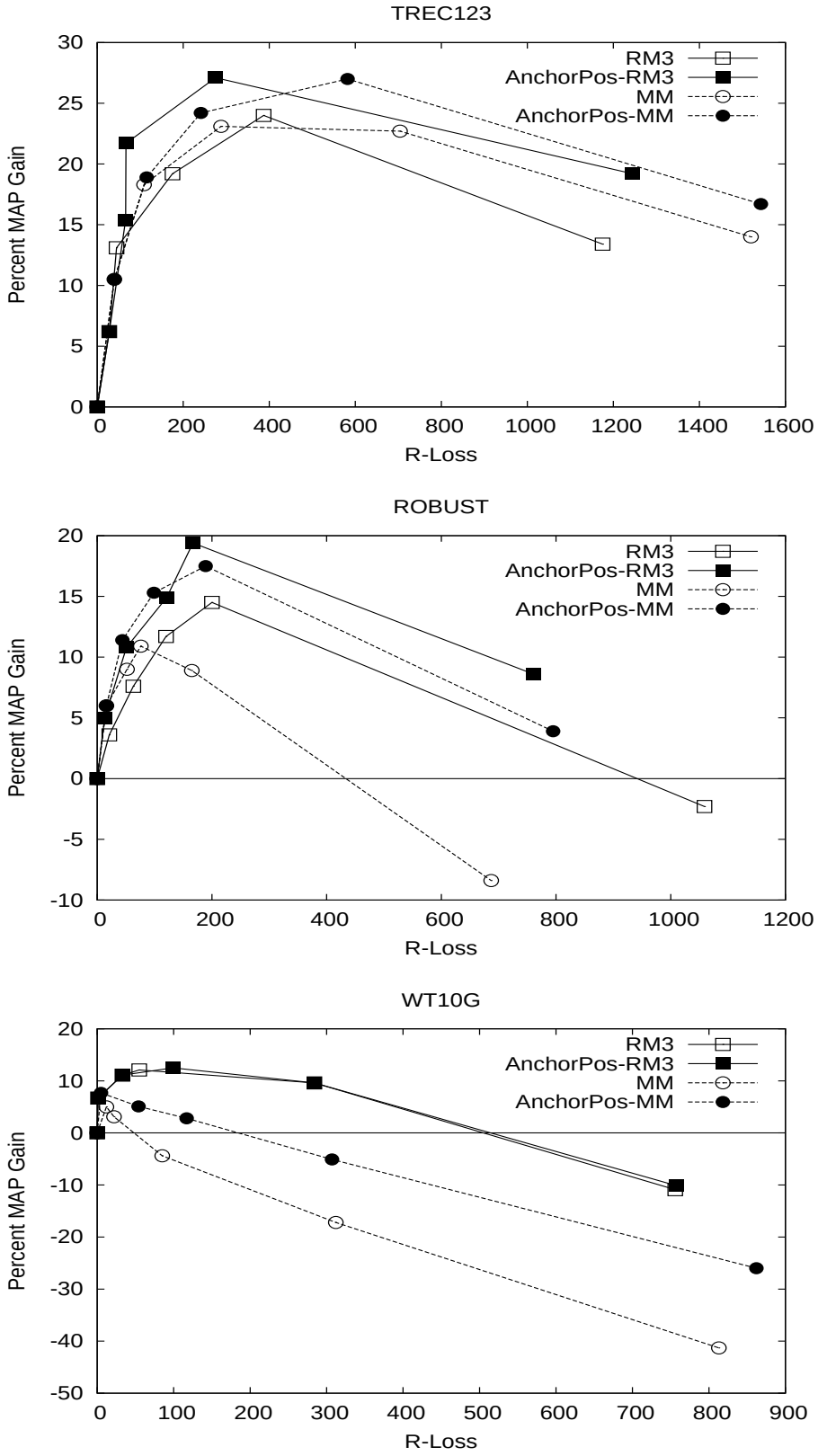

Figure 5.3: MAP risk-reward curves [11]. Note: figures are not to the same scale.

ClipNeg

In the ClipNeg method, the e percent of negative anchor terms that are assigned the highest probability by the negative anchor model are clipped from the given query model. Additional standard clipping is applied by the original probabilities assigned to terms by the query model.

Figure 5.4 contrasts the performance of ClipNeg with that of **ClipRand**;[7] ClipRand clips randomly selected, rather than negative anchor, terms using the exact same approach applied by ClipNeg. The number of terms, ν, assigned with a non-zero probability in the final model is one of the parameters tuned to optimize average retrieval performance. Thus, the performance for $e = 0$ corresponds to optimal standard term clipping — i.e., according to the probabilities they are assigned by the generative query model — while that for $e > 0$ corresponds to optimal combined negative anchor term clipping and standard term clipping. The results are presented for an initial list, $\mathcal{D}_{\text{init}}$, of size $k = 100$.

Figure 5.4 shows that in most cases the performance of ClipNeg increases monotonically with increasing percentage of clipped negative anchor terms. ClipNeg often substantially outperforms ClipRand; the improvements for ROBUST are statistically significant for almost all values of e; the improvements for TREC123 and WT10G are statistically significant for very high values of e except for those for MM over WT10G. Evidently, the performance differences between ClipNeg and ClipRand are smaller for MM than for RM3 as the latter is more effective than the former. A case in point, ClipRand does not improve over standard term clipping ($e = 0$) for RM3, but it does so for MM over WT10G; yet, these improvements are never statistically significant. In contrast, the performance of ClipNeg for $e > 0$ is consistently better than that for $e = 0$; for ROBUST, all improvements are statistically significant; for WT10G many are, while for TREC123 they are not.

All in all, the findings from above further support the merits of clipping negative anchor terms from a query model.

5.3 The discriminative vs. the generative query models

Generative and discriminative query models should be viewed as complementary. Generative models represent the presumed information need and in practice are prone to query drift. The goal of the discriminative query model, on the other hand, is to accurately represent $\mathcal{D}_{\text{init}}$'s ranking, and thus, it can be used to ameliorate potential query drift in generative query models.

To illustrate the differences between the discriminative and generative query models, we provide in Figure 5.5 examples of the query models induced for two queries from the ROBUST dataset. All query models are constructed from a result list, $\mathcal{D}_{\text{init}}$, of 100 documents; the query models were clipped to use 25 terms. The retrieval performance reported is that attained by an optimized interpolation, with a parameter λ or λ_1, of each of the three query models with the original query model (as in Equations 2.8, 2.9 and 3.3). The models resulting from the

[7] In contrast to the evaluation results presented in Tables 5.2, 5.3 and 5.4, where leave-one-out cross validation was used to set free-parameter values, the values of the parameters of all methods considered here are set to optimize MAP over all queries as was the case in the risk-reward analysis above: the goal is to study the potential of clipping negative anchor terms while ameliorating the effects of the generalization, or lack thereof, of effective free-parameter values across queries.

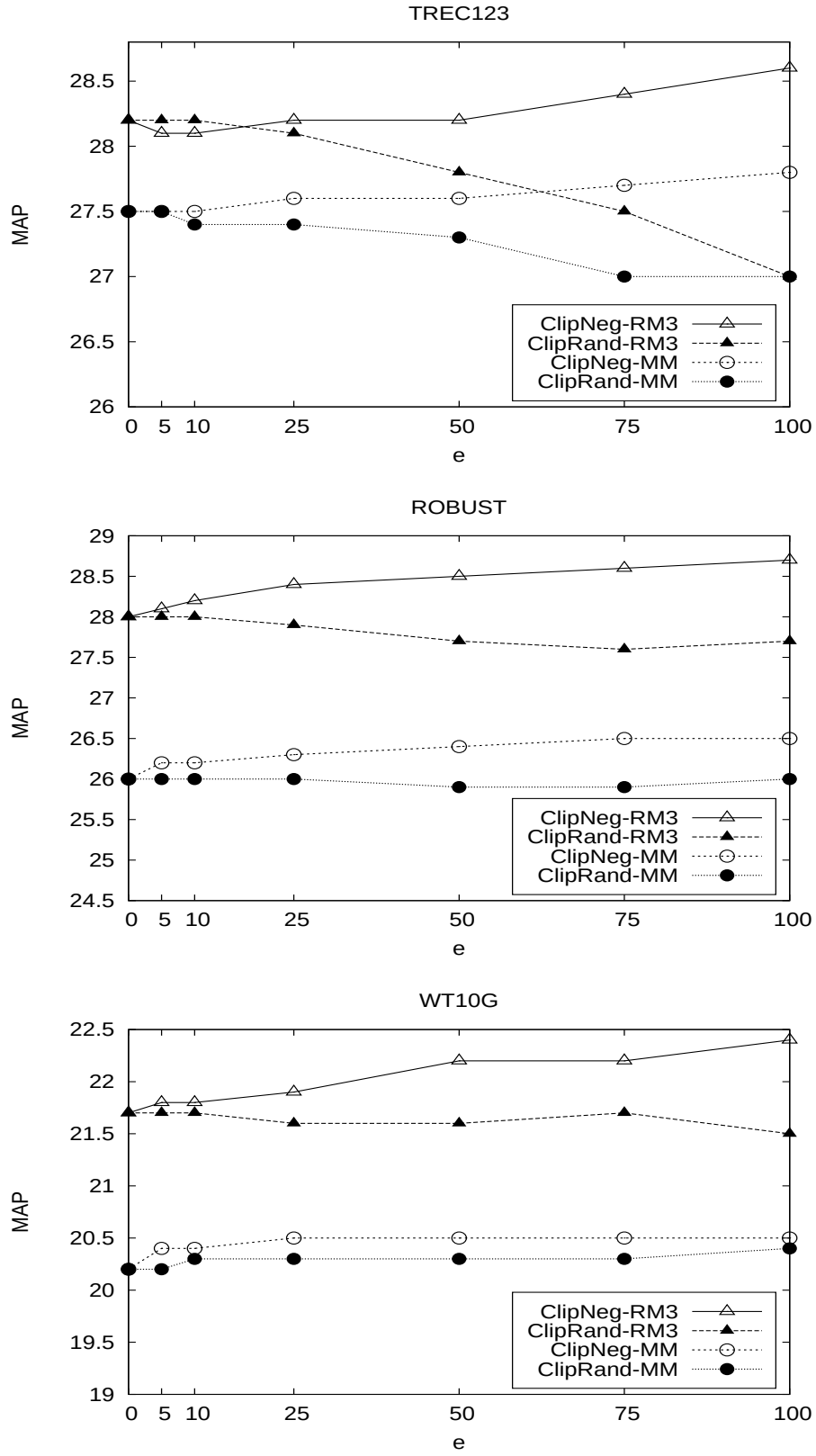

Figure 5.4: The MAP performance of ClipNeg and ClipRand as a function of the percentage of clipped negative anchor terms (e). ClipRand clips randomly selected terms. ClipNeg and ClipRand are applied on RM3 and MM. Initial list, $\mathcal{D}_{\text{init}}$, of 100 documents is used. Note: figures are not to the same scale.

Figure 5.5: Examples of the generative query models (RM1 and θ_T from the mixture model) and the positive anchor model ($\theta_{\vec{w}_+}$). The size of a visualized term is proportional to the probability it is assigned by the query model. The average precision (AP) is the optimal attained by interpolating the query model with the original query model and tuning the interpolation parameter λ (for RM1 and θ_T) and λ_1 (for $\theta_{\vec{w}_+}$). Note: models are not to the same scale.

interpolation are RM3 (based on RM1), MM (based on θ_T) and Q+Pos (based on $\theta_{\vec{w}_+}$; see Section 5.2.3 for details) which interpolates the positive anchor model ($\theta_{\vec{w}_+}$) with the original query model.

Figure 5.5 shows that for both queries, the generative models assign high probabilities to the original query terms or their variants. Indeed, generative query models, as most pseudo-feedback-based query models, reward terms with substantial presence in $\mathcal{D}_{\text{init}}$. As $\mathcal{D}_{\text{init}}$ is retrieved with response to the original query this result is obtained. In contrast, the positive anchor model, $\theta_{\vec{w}_+}$, rewards terms that distinguish high ranked documents from low ranked ones. A case in point, the original query terms are not necessarily positive anchors as can be seen in Figure 5.5. Indeed, if $\mathcal{D}_{\text{init}}$'s ranking is dominated by one of the query terms, the presence of others might have little, or even negative, correlation with the ranking.

We also see in Figure 5.5 that for query #341, the positive anchor model assigns high probability to query related terms (e.g., "terrorist", "baggage" and "passenger") to a more substantial extent than the generative models; and, its retrieval performance is superior. On the other hand, RM1 promotes common (non informative) terms and MM assigns relatively small probabilities to terms that are non included in the original query. The fact that the optimal value of λ for interpolating the generative query models with the original query model is 1 attests to the fact that these two are completely ineffective for query #341, in contrast to the positive anchor model.

For query #308, using all three models improves over the initial ranking, although the positive anchor model is the least effective. However, as noted above, the positive query anchor model is not intended to be a stand-alone query model, but rather used to query anchor the generative query models using the methods presented in Section 3.2.

Next, we provide some statistics (across the three corpora and corresponding query sets) that shed light on the commonalities and differences between the query models. We found that the positive anchor model assigns high probability to terms with a much higher IDF (inverse document frequency) value than that of terms assigned a high probability by the generative query models. For example, we see in Figure 5.5 that the relevance model can reward low IDF terms such as "new" and "make" for query #341. On the other hand, the discriminative query model seeks to differentiate high ranked from low ranked documents in $\mathcal{D}_{\text{init}}$ and is therefore likely to reward high IDF terms. (The importance of using high IDF terms for query expansion has been noted in past work [26, 8, 10].)

Additional finding is that the average number of shared terms among the 25 assigned the highest probability by RM1 and θ_T is 2.33 and 3.5 times higher than that shared by the positive anchor model with RM1 and θ_T, respectively. In other words, the generative models are much more similar to each other, with respect to the terms they promote, than they are to the positive anchor model. This finding provides further support to the complementary nature of the generative models and the discriminative model.

We also found positive Pearson correlation between the prevalence of document pairs in $\mathcal{D}_{\text{init}}$'s wherein a relevant document is ranked higher than a non-relevant one and the retrieval effectiveness (in terms of AP) of using $\theta_{\vec{w}_+}$; specifically, the correlations (all statistically significant at the 95% confidence level) are .52, .39 and .27 for the ROBUST, TREC123 and WT10G corpora

	TREC123			ROBUST			WT10G		
	MAP	p@5	RI	MAP	p@5	RI	MAP	p@5	RI
init	23.0	52.7	–	24.9	47.6	–	19.8	35.5	–
	Relevance Model								
RM3	28.4^{i}	$\mathbf{57.7}^{i}$	48.0	28.2^{i}	48.6	30.9	21.9^{i}	37.7	5.2
Fusion	$27.0^{i,g}$	54.1^{g}	46.7	27.7^{i}	48.9^{i}	39.0	$20.8^{i,g}$	37.9	7.2
ClipNeg	$28.8^{i,g}_{f}$	$\mathbf{57.7}^{i}_{f}$	52.0	$28.9^{i,g}_{f}$	$50.0^{i,g}$	36.5	21.6^{i}	38.1	8.2
AnchorPos	$29.2^{i,g}_{f,c}$	$\mathbf{57.7}^{i}_{f}$	53.3	$29.6^{i,g}_{f}$	$50.1^{i,g}$	31.3	21.8^{i}	37.7	−2.1
ClipNeg+D	$29.5^{i,g}_{f}$	57.6^{i}_{f}	**57.3**	$29.3^{i,g}_{f}$	$50.0^{i,g}$	33.3	$\mathbf{23.1}^{i,g}_{f,*}$	$\mathbf{40.8}^{i,g}_{f,*}$	14.4
AnchorPos+D	$\mathbf{29.9}^{i,g}_{f,*}$	$51.7^{g}_{c,*}$	53.3	$\mathbf{30.2}^{i,g}_{f,c}$	$\mathbf{50.6}^{i}$	**44.2**	$23.0^{i}_{f,*}$	38.6	**20.6**
	Mixture Model								
MM	28.1^{i}	55.1	38.7	27.6^{i}	48.8	25.3	20.8^{i}	36.3	12.4
Fusion	$27.7^{i,g}$	53.6^{g}	39.3	27.8^{i}	48.4	29.7	20.2^{g}	35.1	−3.1
ClipNeg	28.3^{i}	53.3	38.7	27.6^{i}	46.1^{g}_{f}	18.9	21.0^{i}_{f}	33.2^{g}	**16.5**
AnchorPos	$29.1^{i,g}_{f,c}$	**55.9**	40.0	$29.2^{i,g}_{f,c}$	50.0^{i}_{c}	30.9	21.3^{i}_{f}	34.0	9.3
ClipNeg+D	28.1^{i}_{f}	55.1_{f}	38.7	27.6^{i}	48.6_{*}	25.3	20.8^{i}_{f}	33.6^{g}	12.4
AnchorPos+D	$\mathbf{29.7}^{i,g}_{f,c,*}$	$47.9^{i,g}_{f,c,*}$	**46.7**	$\mathbf{29.3}^{i,g}_{f,c}$	$\mathbf{52.3}^{i,g}_{f,c,*}$	**34.9**	$\mathbf{22.3}^{i}$	**38.1**	3.1

Table 5.5: A discriminative model that incorporates the corpus language model. Boldface: best result in a column in a generative model block. Statistically significant differences with the initial ranking (init), generative model (RM3 or MM) and Fusion are marked with 'i', 'g' and 'f', respectively. Statistically significant differences between AnchorPos+D and ClipNeg+D, or AnchorPos and ClipNeg, are marked with 'c'. Statistically significant differences between ClipNeg+D and ClipNeg, or AnchorPos+D and AnchorPos, are marked with '$*$'.

used for evaluation, respectively. This finding provides an additional perspective on the reliance of the discriminative model on the pseudo feedback assumption. That is, the higher the prevalence of documents pairs that are ranked correctly in $\mathcal{D}_{\text{init}}$, the more representative the discriminative model is of the information need.

We do not provide visualization of the negative anchor model ($\theta_{\vec{w}\_}$) as it conveys no additional insight. We found that terms assigned high probability by the negative anchor model often have high IDF values. Presumably, terms with high IDF values occur only in part of the documents in $\mathcal{D}_{\text{init}}$, therefore they are more "explanatory" of the ranking order than terms that occur in most of the documents. These terms can help to differentiate a low ranked document from a high ranked one, as is the case for terms assigned high probability by the positive anchor model. In addition, around 40% of the 25 terms assigned the highest probability by the two generative query models (across the datasets used for evaluation) are negative anchor terms; i.e., they are assigned a non-zero probability by $\theta_{\vec{w}\_}$. Clipping negative anchor terms promoted by the generative models is a method (ClipNeg) we presented in Section 3.2 and whose effectiveness was demonstrated above.

5.4 Utilizing a corpus-based language model

In this section we evaluate the performance of a discriminative query model that explicitly takes into consideration a language model induced form the corpus. (See Section 3.3 for more details.)

5.4.1 Main Result

Table 5.5 presents the effectiveness of our ClipNeg+D and AnchorPos+D methods in comparison to the initial ranking, the generative model on which they are applied (RM3 and MM), the Fusion reference comparison, and the AnchorPos and ClipNeg methods which do not utilize a corpus language model. The performance of the AnchorClip+D method, which integrates ClipNeg+D and AnchorPos+D, is studied below.

Similarly to our previous findings, we see in Table 5.5 that all methods outperform in most cases — often to a statistically significant degree — the initial ranking.

Our AnchorPos+D and ClipNeg+D, when applied on RM3, improve over it in terms of MAP for all three corpora; all improvements are statistically significant, except for WT10G. As for MM, we can see that it is outperformed by AnchorPos+D in terms of MAP for all three corpora; most improvements are statistically significant. The performance of ClipNeg+D in terms of MAP, on the other hand, is equal to that of MM for all corpora.

In terms of p@5, AnchorPos+D improves over the generative model on which it is applied in the majority of relevant comparisons (3 corpora $\times$ 2 generative models), albeit to a statistically significant degree only in one case. While ClipNeg+D significantly outperforms RM3 in terms of p@5 in most corpora, it does not yield any improvements in terms of p@5 over MM.

The RI of ClipNeg+D and AnchorPos+D is in all cases higher than that of RM3. The RI of AnchorPos+D is in most cases higher than that of MM, and the RI of ClipNeg+D is equal to that of MM.

We conclude, based on the above results, that ClipNeg+D is not effective when applied to MM. A possible explanation is that MM also explicitly utilizes the corpus in its estimation. Consequently, frequent terms in the corpus may receive low probabilities in the model [10]. Thus, frequent terms are likely to be clipped using the standard term clipping. ClipNeg+D, on the other hand, performs term clipping using $\theta^D_{\vec{w}_-}$. Since terms in $\theta^D_{\vec{w}_-}$ are also likely to have high frequency in the corpus, the merit in clipping these on top of standard clipping may be limited.

We also see in Table 5.5 that using positive anchor terms (AnchorPos+D) is almost always more effective in terms of MAP than using negative anchor terms (ClipNeg+D), regardless to the generative query model. In terms of p@5, neither AnchorPos+D nor ClipNeg+D dominates the other. However, it is important to highlight that the performance in terms of MAP of ClipNeg+D when applied on RM3 is similar to that of AnchorPos+D for all three corpora, and the differences between them are never statistically significant.

The ClipNeg+D and AnchorPos+D methods outperform the Fusion reference comparison in terms of MAP and p@5 in a vast majority of cases; all the improvements in terms of MAP are statistically significant. In terms of RI the performance of Fusion is in all cases inferior to that of AnchorPos+D and in most cases inferior to that of ClipNeg+D.

Next, we turn to compare the performance of methods that utilize the corpus model to those that do not. We can see that AnchorPos+D outperforms AnchorPos in a majority of relevant comparisons: 10 comparisons out of 12 (2 generative models $\times$ 3 corpora $\times$ 2 evaluation measures); some of the improvements are statistically significant. Moreover, AnchorPos+D posts more improvements over the generative model on which it is applied (RM3 and MM), than AnchorPos does; the number of significant improvements, however, is equal for both methods.

Specifically, while AnchorPos fails to improve (MAP) over RM3 in WT10G, AnchorPos+D improves over both generative models for all three corpora. In terms of RI, AnchorPos+D also outperforms AnchorPos in a vast majority of cases.

ClipNeg+D outperforms ClipNeg in most relevant comparisons (3 corpora $\times$ 2 evaluation measures) when applied on RM3; the improvements in the case of WT10G are statistically significant. Moreover, ClipNeg+D outperforms ClipNeg in terms of RI in most corpora when applied on RM3. When ClipNeg+D is applied on MM it is being outperformed by ClipNeg in terms of MAP, and it outperforms ClipNeg in terms of p@5 in most cases. However, only one difference between these methods is statistically significant. In addition, ClipNeg+D outperforms ClipNeg in terms of RI only in one corpus. The findings in the case of MM can be attributed to the general ineffectiveness of ClipNeg+D when applied on it. Furthermore, in the case of RM3, we can see that in contrast to ClipNeg, ClipNeg+D is highly effective, outperforming it in a majority of relevant comparisons (3 corpora $\times$ 3 evaluation measures); all MAP and p@5 improvements are statistically significant.

As we see in Table 5.5, the AnchorPos and ClipNeg methods in the case of WT10G are not as effective as in other corpora. As mentioned in Section 5.2, this difference in performance can be attributed to the poor quality of the initial rankings of queries in this corpus. However, when the corpus is explicitly incorporated into the model, we can see that the resulting methods can be quite effective for WT10G. Both AnchorPos+D and ClipNeg+D outperform the generative models in most cases by a relatively large margin. However, only one improvement is statistically significant. The lack of significant improvements in WT10G can be due to its relatively small number of queries. Moreover, the methods are also very robust (in terms of RI). The RI of AnchorPos+D, for instance, is almost four times higher than that of RM3.

All in all, we can conclude that the incorporation of the corpus in the discriminative query model results in more effective methods.

AnchorClip+D

The AnchorClip+D method integrates the ClipNeg+D and AnchorPos+D methods in the same way AnchorClip integrates ClipNeg and AnchorPos. Table 5.6 presents a performance comparison of AnchorClip+D with ClipNeg+D and AnchorPos+D.

We see that AnchorClip+D outperforms ClipNeg+D in terms of MAP in most relevant comparisons (2 generative models $\times$ 3 corpora); all of the improvements are statistically significant. In terms of p@5 AnchorClip+D is outperformed by ClipNeg+D in most cases; half of the differences are statistically significant.

When comparing AnchorClip+D and AnchorPos+D, we see that AnchorClip+D dominates AnchorPos+D (MAP and p@5). However, these differences are not statistically significant in vast majority of cases.

In addition, we can see that AnchorClip+D outperforms AnchorClip in a majority of relevant comparisons (3 corpora $\times$ 2 generative models $\times$ 2 evaluation measures); most of the differences are statistically significant. This finding comes as no surprise as we saw earlier in Table 5.5 that the methods which explicitly incorporate the corpus are more effective than those that do not.

	TREC123			ROBUST			WT10G		
	MAP	p@5	RI	MAP	p@5	RI	MAP	p@5	RI
Relevance Model									
ClipNeg	28.8	**57.7**	52.0	28.9	50.0	36.5	21.6	38.1	8.2
AnchorPos	29.2^{c}	**57.7**	53.3	29.6	50.1	31.3	21.8	37.7	−2.1
AnchorClip	29.4^{c}	56.7	52.0	29.5	**51.1**	34.1	21.1	37.3	2.1
ClipNeg+D	29.5	57.6	**57.3**	29.3	50.0	33.3	$\mathbf{23.1}_{*}$	$\mathbf{40.8}_{*}$	14.4
AnchorPos+D	29.9_{*}	51.7^{c}_{*}	53.3	30.2^{c}	50.6	44.2	23.0_{*}	38.6	**20.6**
AnchorClip+D	$\mathbf{30.0}^{c}_{*}$	57.2^{a}	54.0	$\mathbf{30.3}^{c}_{*}$	$45.9^{c,a}_{*}$	**48.2**	$\mathbf{23.1}_{*}$	$\mathbf{40.8}_{*}$	**20.6**
Mixture Model									
ClipNeg	28.3	53.3	38.7	27.6	46.1	18.9	21.0	33.2	**16.5**
AnchorPos	29.1^{c}	55.9	40.0	29.2^{c}	50.0^{c}	30.9	21.3	34.0	9.3
AnchorClip	29.4^{c}	$\mathbf{56.4}^{c}$	38.7	29.1^{c}	50.4^{c}	28.1	21.5	32.6	8.2
ClipNeg+D	28.1	55.1	38.7	27.6	48.6_{*}	25.3	20.8	33.6	12.4
AnchorPos+D	29.7^{c}_{*}	47.9^{c}_{*}	**46.7**	29.3^{c}	$\mathbf{52.3}^{c}_{*}$	**34.9**	**22.3**	38.1	3.1
AnchorClip+D	$\mathbf{29.8}^{c}$	$51.9^{c,a}_{*}$	**46.7**	$\mathbf{30.1}^{c,a}_{*}$	$\mathbf{52.3}^{c}_{*}$	**34.9**	21.3	$\mathbf{38.8}^{c}$	3.1

Table 5.6: Comparison of AnchorClip+D with ClipNeg+D and AnchorPos+D. The best result in a column in a generative model block is boldfaced. Statistically significant differences with ClipNeg+D (or ClipNeg) and AnchorPos+D (or AnchorPos) are marked with 'c' and 'a', respectively. Statistically significant differences between ClipNeg+D and ClipNeg, AnchorPos+D and AnchorPos, or AnchorClip+D and AnchorClip, are marked with '$*$'. Note: we use 'c' and 'a' to mark statistically significant differences only between methods in the same block of methods.

Our conclusion here, similarly to the original model (AnchorClip), is that there is no clear merit in combining ClipNeg+D and AnchorPos+D.

5.4.2 Further Analysis

In this section we further explore the utilization of positive anchor terms in the model that utilizes the corpus language model (the AnchorPos+D method). Setting $\lambda_2 = 0$ in AnchorPos+D yields a query model, **Q+Pos+D**, that interpolates the original query model with the clipped positive anchor model ($\theta^{D}_{\vec{w}_+}$); the generative query model is not used. Setting $\lambda_1 = 0$ results in the **M+Pos+D** method which integrates the positive anchor model with the generative query model; term clipping is applied but not interpolation with the original query model. Table 5.7 presents the performance of these specific cases of AnchorPos+D.

In two out of three corpora, Q+Pos+D outperforms the initial ranking (MAP and p@5); the map differences are statistically significant. Q+Pos+D is often outperformed by the generative query model (RM3 and MM). However, in most cases the differences are not statistically significant. More specifically, The MAP of Q+Pos+D is often close to or higher than that of RM3 (the most effective generative model studied in this work).

Q+Pos+D outperforms Q+Pos in terms of MAP in two of the corpora. The improvements are statistically significant. In terms of p@5, Q+Pos+D is outperformed by Q+Pos for all corpora. However, only in one case the difference is statistically significant.

In terms of MAP, the performance of M+Pos+D is often higher than that of the initial ranking and the generative query model, except for WT10G; half of the improvements are statistically significant. According to this finding, we can conclude that in some cases direct query anchoring (using the original query) can be replaced by our novel indirect query anchoring approach, resulting in better performance. Moreover, M+Pos+D dominates M+Pos for all reference comparisons, except for one case; half of the improvements in terms of MAP and p@5

	TREC123			ROBUST			WT10G		
	MAP	p@5	RI	MAP	p@5	RI	MAP	p@5	RI
init	23.0	52.7	–	24.9	47.6	–	19.8	35.5	–
Relevance Model									
RM3	28.4^{i}	$\mathbf{57.7}^{i}$	48.0	28.2^{i}	48.6	30.9	21.9^{i}	37.7	5.2
AnchorPos	$29.2^{i,g}$	$\mathbf{57.7}^{i}$	**53.3**	$29.6^{i,g}$	$50.1^{i,g}$	31.3	21.8^{i}	37.7	−2.1
Q+Pos	$25.3^{i,g}_{a}$	55.3^{i}	28.7	$26.6^{i,g}_{a}$	49.7^{i}	4.8	20.4^{g}_{a}	36.3	10.3
M+Pos	$27.4^{i}_{a,p}$	56.8^{i}	17.3	27.0^{i}_{a}	$43.9^{i,g}_{a,p}$	3.6	$15.5^{i,g}_{a,p}$	34.6	−50.5
AnchorPos+D	$\mathbf{29.9}^{i,g}_{*}$	51.7^{g}_{*}	**53.3**	$\mathbf{30.2}^{i,g}$	50.6^{i}	**44.2**	$\mathbf{23.0}^{i}_{*}$	**38.6**	**20.6**
Q+Pos+D	$28.2^{i}_{a,*}$	53.1^{g}_{*}	42.7	$28.7^{i}_{a,*}$	49.2_{a}	32.1	19.6^{g}_{a}	34.8	4.1
M+Pos+D	$29.1^{i}_{a,*}$	52.7_{*}	36.0	$29.3^{i,g}_{*}$	$\mathbf{50.9}^{i}_{p,*}$	13.3	$19.3^{g}_{a,*}$	37.1	−32.0
Mixture Model									
MM	28.1^{i}	55.1	38.7	27.6^{i}	48.8	25.3	20.8^{i}	36.3	**12.4**
AnchorPos	$29.1^{i,g}$	**55.9**	40.0	$29.2^{i,g}$	50.0^{i}	**30.9**	21.3^{i}	34.0	9.3
Q+Pos	$25.3^{i,g}_{a}$	55.3^{i}	28.7	26.6^{i}_{a}	49.7^{i}	4.8	20.4	36.3	10.3
M+Pos	$26.8^{i,g}_{a}$	$48.4^{i,g}_{a,p}$	2.7	25.9^{g}_{a}	$45.9^{g}_{a,p}$	−1.2	$14.7^{i,g}_{a,p}$	$30.3^{i,g}_{a}$	−45.4
AnchorPos+D	$\mathbf{29.7}^{i,g}_{*}$	$47.9^{i,g}_{*}$	**46.7**	$\mathbf{29.3}^{i,g}$	$\mathbf{52.3}^{i,g}_{*}$	**34.9**	$\mathbf{22.3}^{i}$	**38.1**	3.1
Q+Pos+D	$28.2^{i}_{a,*}$	$53.1_{a,*}$	42.7	28.7^{i}_{*}	49.2_{a}	32.1	19.6_{a}	34.8	4.1
M+Pos+D	$28.4^{i}_{a,*}$	49.6^{g}	24.0	$29.0^{i,g}_{*}$	47.6_{a}	16.9	$15.7^{i,g}_{a,p}$	$30.9^{i,g}_{a,p}$	−42.3

Table 5.7: AnchorPos+D and its two specific cases: Q+Pos+D and M+Pos+D. The best result in a column in a generative model block is boldfaced. Statistically significant differences with the initial ranking, the generative query model (RM3 or MM), AnchorPos+D (or AnchorPos), and M+Pos+D (or M+Pos) are marked with 'i', 'g', 'a' and 'p', respectively. Note: we use 'a' and 'p' to mark statistically significant differences only between methods in the same block of methods.

are statistically significant. This further attests to the merit of explicitly incorporating the corpus in the discriminative model.

5.5 Differential weighting of pairs in SVMrank

The underlying assumption of the discriminative query model is that for *any* pair of documents, the higher ranked document is more likely to be relevant than the lower ranked one. The assumption entails that all pairwise preferences, generated using the original ranking, are equally important in the estimation of the model, regardless of the actual ranks of the documents in the pair. Since the initial result list was retrieved using the original query, this assumption may not necessarily hold in reality. For example, the difference in relevance likelihood between two documents placed in the first and second ranks may not be the same as the difference between two documents placed in the lowest ranks.

In order to challenge this assumption we experimented with an SVMrank approach that assigns weight for every pair of documents in the ranking [6]. Specifically, every pair was scored using the difference in DCG (discounted cumulative gain) of the individual documents. Given two documents with the ranks i and j in the initial result list, where $j > i$, the score assigned to the pair is:

$$\Delta DCG(i,j) = \frac{2^{rel_i} - 1}{\log i + 1} - \frac{2^{rel_j} - 1}{\log j + 1}; \tag{5.1}$$

where rel_i and rel_j are the relevance scores of the documents i and j, respectively. Assuming that the higher ranked document in the pair is more relevant than the lower ranked one, we set rel_i and rel_j to 2 and 1, respectively.

	TREC123			ROBUST			WT10G		
	MAP	p@5	RI	MAP	p@5	RI	MAP	p@5	RI
RM3	28.4	**57.7**	48.0	28.2	48.6	30.9	**21.9**	**37.7**	**5.2**
AnchorPos	**29.2**^{g}	**57.7**	**53.3**	29.6^{g}	**50.1**^{g}	31.3	21.8	**37.7**	−2.1
WeightedAnchorPos	**29.2**^{g}	55.5$^{g}_{a}$	**53.3**	**29.7**^{g}	46.1$^{g}_{a}$	**35.3**	21.0$^{g}_{a}$	35.7$^{g}_{a}$	−6.2

Table 5.8: Differential weighting of pairs in the discriminative model. The best result in a column in a generative model block is boldfaced. Statistically significant differences with RM3 and AnchorPos are marked with 'g' and 'a', respectively.

Results of preliminary experiments with this version of the model are presented in Table 5.8. Specifically, we evaluated the performance of AnchorPos, which uses this version of the discriminative query model (denoted WeightedAnchorPos), when applied to RM3. We can see in Table 5.8 that WeightedAnchorPos never outperforms AnchorPos to a statistically significant degree. Based on these results, we conclude that differential weighting of pairs in the model has no clear merit for indirect query anchoring using the positive anchor model.

Discussion

Direct query anchoring was shown in previous work to be a key component of query models for ameliorating potential query drift (e.g., [1, 42]). In this work we devised indirect query anchoring approaches that are complementary to past approaches. Specifically, we compared our approaches with two common and effective query anchoring techniques: interpolation with a model of the original query, and term clipping.

Our approaches differ from the existing ones in the manner in which they modify the probability distribution, defined by the query model. The probability mass of the original query terms in the distribution is increased via interpolation with a model of the original query. In term clipping, the probability mass of the terms with the highest probabilities is further increased. And, in our approaches the probability mass of positive anchor terms is increased by boosting their probabilities (AnchorPos) or by clipping negative anchor terms (ClipNeg).

Empirically, we showed that applying our approaches on top of query models that already apply the above mentioned techniques yields further improvements. Nevertheless, we demonstrated that the effectiveness of our approaches as stand-alone query anchoring techniques (M+Pos) is still inferior to that of the direct approaches. Thus, direct query anchoring is still an important technique for mitigating the risk of query drift.

Chapter 6

Conclusions and Future Work

In this thesis we addressed the ad hoc document retrieval task: ranking documents in a corpus by their relevance to an information need expressed by a query. One of the challenges in the ad hoc retrieval task is modeling the presumed information need of the user. To that end, query models are often induced using the original query. However, such query models often lack information due to the relatively short queries of the users. To address this issue, techniques for inducing query models from pseudo relevant documents have been proposed.

We presented a novel unsupervised pseudo-feedback-based discriminative query model. The model is induced from an initially retrieved list using a learning-to-rank-approach by considering pairwise document preferences induced from the list. More specifically, for each pair of documents in the list, the one ranked higher is considered more relevant than that ranked lower. The resultant query model constitutes a term-based representation of the list ranking.

We demonstrated the empirical merits of methods using the discriminative model to query anchor highly effective generative query models: emphasizing terms that are positively correlated with the initial ranking, and clipping terms that are negatively correlated with the initial ranking. We showed that there is no clear merit in combining the two approaches.

We empirically demonstrated the complementary nature of generative and discriminative query models. This finding further supports the motivation in developing methods that integrate them. In this work we devised several methods that perform the integration at the model level. In future work other types of methods that utilize the discriminative model can be examined.

We suggest several possible directions for future work. Using additional learning-to-rank methods to induce a term-based representation from a retrieved list. Furthermore, studying the effectiveness of the discriminative query model when applied on additional pseudo-feedback-based query models is another future direction. Finally, utilizing the discriminative query model in other tasks of information retrieval is also a venue we intend to explore.

References

[1] Nasreen Abdul-Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Marck D. Smucker, and Courtney Wade. UMASS at TREC 2004 — novelty and hard. In *Proc. of TREC-13*, 2004.

[2] Giambattista Amati, Claudio Carpineto, and Giovanni Romano. Query difficulty, robustness, and selective application of query expansion. In *Proc. of ECIR*, pages 127–137, 2004.

[3] Bodo Billerbeck and Justin Zobel. When query expansion fails. In *Proceedings of SIGIR*, pages 387–388, 2003.

[4] Chris Buckley, Gerard Salton, James Allan, and Amit Singhal. Automatic query expansion using SMART: TREC3. In *Proc. of TREC-3*, pages 69–80, 1994.

[5] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proc. of SIGIR*, pages 243–250, 2008.

[6] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. Adapting ranking svm to document retrieval. In *Proceedings of ACM SIGIR*, pages 186–193. ACM, 2006.

[7] Claudio Carpineto, Renato de Mori, Giovanni Romano, and Brigitte Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, 19(1):1–27, 2001.

[8] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys*, 44(1):1, 2012.

[9] Stéphane Clinchant and Éric Gaussier. Information-based models for ad hoc IR. In *Proc. of SIGIR*, pages 234–241, 2010.

[10] Stéphane Clinchant and Éric Gaussier. A theoretical analysis of pseudo-relevance feedback models. In *Proc. of ICTIR*, 2013.

[11] Kevyn Collins-Thompson. Reducing the risk of query expansion via robust constrained optimization. In *Proceedings of CIKM*, pages 837–846, 2009.

[12] Kevyn Collins-Thompson and Jamie Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proc. of SIGIR*, pages 303–310, 2007.

[13] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. A language modeling framework for selective query expansion. Technical Report IR-338, Center for Intelligent Information Retrieval, University of Massachusetts, 2004.

[14] Fernando Diaz and Donald Metzler. Improving the estimation of relevance models using large external corpora. In *Proc. of SIGIR*, pages 154–161, 2006.

[15] Edward A. Fox and Joseph A. Shaw. Combination of multiple searches. In *Proc. of TREC-2*, 1994.

[16] Donna Harman. Relevance feedback revisited. In *Proc. of SIGIR*, pages 1–10, 1992.

[17] Ben He and Iadh Ounis. Finding good feedback documents. In *Proc. of CIKM*, pages 2011–2014, 2009.

[18] Ben He and Iadh Ounis. Studying query expansion effectiveness. In *Proc. of ECIR*, pages 611–619, 2009.

[19] Frederick Jelinek and Robert L Mercer. Interpolated estimation of markov source parameters from sparse data.

[20] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proc. of SIGKDD*, pages 133–142, 2002.

[21] Mostafa Keikha, Jangwon Seo, W. Bruce Croft, and Fabio Crestani. Predicting document effectiveness in pseudo relevance feedback. In *Proc. of CIKM*, pages 2061–2064, 2011.

[22] John D. Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proc. of SIGIR*, pages 111–119, 2001.

[23] Victor Lavrenko and W. Bruce Croft. Relevance-based language models. In *Proc. of SIGIR*, pages 120–127, 2001.

[24] Kyung-Soon Lee, W. Bruce Croft, and James Allan. A cluster-based resampling method for pseudo-relevance feedback. In *Proc. of SIGIR*, pages 235–242, 2008.

[25] Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3), 2009.

[26] Yuanhua Lv and ChengXiang Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In *Proc. of CIKM*, pages 1895–1898, 2009.

[27] Yuanhua Lv and ChenXiang Zhai. Adaptive relevance feedback in information retrieval. In *Proc. of CIKM*, pages 255–264, 2009.

[28] Mandar Mitra, Amit Singhal, and Chris Buckley. Improving automatic query expansion. In *Proc. of SIGIR*, pages 206–214, 1998.

[29] Joseph John Rocchio. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.

[30] Tetsuya Sakai, Toshihiko Manabe, and Makoto Koyama. Flexible pseudo-relevance feedback via selective sampling. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(2):111–135, 2005.

[31] Gerard Salton. Automatic information organization and retrieval. 1968.

[32] Jerard Salton, Anita Wong, and Chung Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[33] Jangwon Seo and W. Bruce Croft. Geometric representations for multiple documents. In *Proc. of SIGIR*, pages 251–258, 2010.

[34] Fei Song and W. Bruce Croft. A general language model for information retrieval. In *Proc. of SIGIR*, pages 279–280, 1999.

[35] Natali Soskin, Oren Kurland, and Carmel Domshlak. Navigating in the dark: Modeling uncertainty in ad hoc retrieval using multiple relevance models. In *Proc. of ICTIR*, pages 79–91, 2009.

[36] Amanda Spink and Bernard J Jansen. A study of web search trends. *Webology*, 1(2):4, 2004.

[37] Tao Tao and ChengXiang Zhai. A mixture clustering model for pseudo feedback in information retrieval. In *Proc. of IFCS*, pages 541–552, 2004. Invited paper.

[38] Tao Tao and ChengXiang Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proc. of SIGIR*, pages 162–169, 2006.

[39] Raghavendra Udupa, Abhijit Bhole, and Pushpak Bhattacharyya. "A term is known by the company it keeps": On selecting a good expansion set in pseudo-relevance feedback. In *Proc. of ICTIR*, pages 104–115, 2009.

[40] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proc. of SIGIR*, pages 4–11, 1996.

[41] Zheng Ye, Ben He, Xiangji Huang, and Hongfei Lin. Revisiting Rocchio's relevance feedback algorithm for probabilistic models. In *Proc. of AIRS*, pages 151–161, 2010.

[42] Chengxiang Zhai and John D. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proc. of CIKM*, pages 403–410, 2001.

[43] Chengxiang Zhai and John D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proc. of SIGIR*, pages 334–342, 2001.

[44] Liron Zighelnic and Oren Kurland. Query-drift prevention for robust query expansion. In *Proc. of SIGIR*, pages 825–826, 2008.

מודלי שאילתא מבחינים (דיסקרימינטיביים)

סער קוזי

מודלי שאילתא מבחינים (דיסקרימינטיביים)

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר

מגיסטר למדעים בהנדסת ניהול מידע

סער קוזי

הוגש לסנט הטכניון – מכון טכנולוגי לישראל

אדר התשע"ז חיפה פברואר 2017

המחקר בוצע בהנחייתו של פרופסור אורן קורלנד בפקולטה להנדסת תעשייה וניהול. חלק מן התוצאות בחיבור זה פורסמו כמאמר מאת המחבר ושותפים למחקר בכנס במהלך תקופת המחקר בלימודי מוסמכים של המחבר, אשר גרסתו העדכנית ביותר היא:

Saar Kuzi, Anna Shtok, and Oren Kurland. Query anchoring using discriminative query models. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, pages 219-228. ACM, 2016.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

אני מודה לארווין וג'ואן ג'ייקובס על המלגה הנדיבה.

תקציר

מנועי חיפוש יעילים הינם בעלי ערך רב כיום לאור העלייה הדרמטית בשנים האחרונות בכמויות המידע הדיגיטלי הזמין למשתמשים היות והם מקלים על מציאה של מידע רלוונטי במאגרים גדולים של מידע דיגיטלי. מאגרים אלו מורכבים, למשל, ממסמכים, קטעי וידאו, או תמונות. משימת *האחזור האד הוקי* (ad hoc retrieval), המבוצעת על ידי מנועי חיפוש, הינה במוקד של עבודת מחקר זאת. המטרה היא לאחזר מסמכים הרלוונטיים לצורך במידע של המשתמש כפי שבא לידי ביטוי בשאילתא מתוך מאגר של מסמכים [31]. אחד האתגרים של משימת האחזור האד הוקי הוא מידול של הצורך במידע. לטובת כך, פותחו מודלי שאילתא שונים אשר במקרים רבים נבנים תוך שימוש בשאילתא המקורית (למשל, [32,34]).

שאילתות, בייחוד במנועי חיפוש ברשת האינטרנט, הן קצרות בממוצע [36]. לפיכך, הן לעיתים לא מייצגות נאמנה את הצורך במידע של המשתמש. לדוגמא, ייתכן פער בין אוצר המילים של השאילתא הקצרה לבין זה של המסמכים הרלוונטיים. כלומר, ייתכנו מסמכים רלוונטיים המכילים רק חלק, או אף אחת, ממילות השאילתא. למשל, אם השאילתא של המשתמש היא "רכב" ומסמך רלוונטי מנגד משתמש במונח "מכונית". לפיכך, שימוש במודלי שאילתא המסתמכים רק על מילות השאילתא המקורית עלול להביא לפגיעה בביצועי האחזור.

מספר מודלי שאילתא פותחו במטרה לייצג את הצורך במידע באופן יעיל יותר מאשר מודלי שאילתא המבוססים על השאילתא הקצרה בלבד. מודלי שאילתא אלו, למשל, יכולים לנקוט בגישה של הרחבת שאילתא [8]. באמצעות גישה זו מודלי שאילתא מגשרים על הפער הלקסיקלי בין השאילתא למסמכים הרלוונטיים על ידי הוספת מילים אשר ככל הנראה קשורות לצורך במידע של המשתמש. יתרה מכך, שימוש במודלי שאילתא יכול לשפר את ביצועי האחזור על ידי מתן חשיבות גבוהה למילים אשר בסבירות גבוהה יכולות להבחין בין מסמכים רלוונטיים ללא רלוונטיים, וחשיבות נמוכה למילים אשר אינן אפקטיביות בביצוע הבחנה זאת (לדוגמא, [23,1,42]).

מודלי שאילתא יכולים להיבנות תוך שימוש ברשימת תוצאות ראשונית של מסמכים המדורגים גבוה ביותר על ידי חיפוש ראשוני שבוצע ביחס לשאילתא המקורית. לדוגמא, ישנם מודלים אשר עושים שימוש במשוב מהמשתמש על המסמכים ברשימת התוצאות. מודלים אלו, למשל, יכולים להגביר את חשיבותן של מילים השכיחות במסמכים הרלוונטיים, ולהפחית את חשיבותן של מילים השכיחות במסמכים הלא רלוונטיים [29]. במקרים בהם משוב ישיר מן המשתמש אינו זמין, ניתן לעשות שימוש בפסאודו משוב (pseudo feedback) [8]. גישות המבוססות על פסאודו משוב מתייחסות למסמכים המצויים בדירוגים הגבוהים ברשימה ההתחלתית כרלוונטיים לצורך במידע. הנחת הפסאודו משוב היא שככל שמסמך מדורג גבוה יותר כך הסבירות שהוא רלוונטי עולה.

מסמכים ברשימת התוצאות הראשונית יכולים להיות לא רלוונטיים, ומסמכים רלוונטיים יכולים להכיל מילים שאינן קשורות לשאילתא [16,18,27]. כתוצאה מכך, מודל השאילתא אשר נבנה תוך שימוש ברשימה זו יכול לזלוג מהצורך במידע של המשתמש [28]; דהיינו, המודל יכול לייחס חשיבות להיבטים שאינם קשורים לצורך במידע. לאור זאת, שימוש במודל השאילתא יכול להוביל לירידה בביצועי האחזור, לעיתים עד כדי כך ששימוש רק בשאילתא המקורית הינו יעיל יותר

[2,13]. מספר גישות, ידועות בשם *עוגן שאילתא*, הוצעו על מנת להפחית את הסיכון שבהסתמכות על פסאודו משוב. גישות עוגן שאילתא ישירות מבוססות על שימוש בשאילתא המקורית כעוגן כשאר נעשה שימוש בפסאודו משוב. לדוגמא, אינטרפולציה של מודל השאילתא עם מודל של השאילתא המקורית הינה גישת עוגן שאילתא ישירה נפוצה (למשל, [29,7,1,43,26,9]).

גישות עוגן שאילתא עקיפות מבוססות על מספר הנחות ביחס לפסאודו משוב ולקשר שלו לצורך במידע. לדוגמא, קטימת מודל השאילתא על ידי שימוש רק במילים עם החשיבות הגבוהה ביותר במודל הינה גישה נפוצה (למשל, [4,40,43,1,41]). ההנחה היא שמילים אלו בסבירות גבוהה מייצגות את הצורך במידע שהרי הן מייצגות את רשימת התוצאות. לפי גישה עקיפה נוספת ניתנת חשיבות גבוהה יותר למילים המופיעות במסמכים בדירוגים גבוהים ברשימה התחלתית, לעומת מילים המופיעות במסמכים המדורגים נמוך יותר ברשימה [23,1,37,33]. ההנחה היא שככל שמסמך מדורג גבוה יותר, כך גוברת הסבירות שהוא רלוונטי היות ורשימת התוצאות הראשונית נוצרה תוך שימוש בשאילתא המקורית.

בעבודה זו אנו מציגים גישת עוגן שאילתא עקיפה חדשה אשר יכולה לשמש מודלי שאילתא קיימים. הגישה עושה שימוש במודל דיסקרמינטיבי לא מונחה (unsupervised) המבוסס על פסאודו משוב ואשר נבנה מרשימת התוצאות הראשונית. מודל זה הינו חדש לעבודה זאת. המודל משמש כייצוג דיסקרמינטיבי מדויק ברמת המילים של הדירוג של רשימת התוצאות הראשונית. ככזה, המודל יכול לשמש באמצעות שיטות שאנו מציעים, כעוגן שאילתא.

על מנת לבנות את המודל הדיסקרמינטיבי, אנו ממנפים את הנחת הפסאודו משוב באופן הבא. אנו מניחים כי לכל זוג מסמכים ברשימה הראשונית, המסמך בדירוג הגבוה הינו רלוונטי יותר מהמסמך בדירוג הנמוך. בהסתמך על הנחה זאת, אנו יוצרים קבוצה של העדפות בין זוגות מסמכים תוך שימוש בכל הזוגות הסדורים של מסמכים בדירוג הראשוני. לאחר מכן, העדפות אלו משמשות כקבוצת אימון עבור אלגוריתם ללמידת דירוג (learning-to-rank algorithm); בעבודה זאת נעשה שימוש ב- SVMrank [20]. אנו מבצעים התאמה מדויקת (overfitting) של המודל הנלמד לדירוג של רשימת התוצאות הראשונית על מנת לקבל ייצוג דיסקרמינטיבי, מדויק ככל הניתן, ברמת המילים של הדירוג. כתוצאה מכך, אנו מקבלים מודל שאילתא דיסקרמינטיבי המשקף את ההטלה של השאילתא על מאגר המסמכים כפי שבא לידי ביטוי ברשימת התוצאות.

המודל המתקבל מורכב משתי קבוצות של מילים: בעלות מתאם חיובי או שלילי עם הדירוג הראשוני. מילים בעלות מתאם חיובי הינן בעלות נטייה להופיע יותר במסמכים בדירוג גבוה לעומת נמוך, וההפך עבור מילים בעלות מתאם שלילי. ההנחה היא כי מילים בעלות מתאם חזק (חיובי או שלילי) עם הדירוג הראשוני יכולות להבחין בצורה אפקטיבית בין מסמכים רלוונטיים ולא רלוונטיים. אנו מתייחסים למילים עם מתאם חיובי ושלילי כעוגנים חיוביים ושליליים, בהתאמה.

תוך שימוש במודל הדיסקרמינטיבי, אנו מפתחים שיטות עוגן שאילתא עבור מודלי שאילתא קיימים. לדוגמא, מילים בעלות חשיבות גבוהה במודל שאילתא ושהינן עוגנים חיוביים במידה רבה צריכות לקבל חשיבות יתרה. לעומת זאת, החשיבות של מילות עוגן שלילי במודל השאילתא צריכה לקטון. בנוסף, אנו מציגים גישה אשר משתמשת בעוגנים חיוביים ושליליים. אנו מדגימים

באמצעות ניסויים את הערך המוסף בהפעלת הגישות על שני מודלי שאילתא גנרטיביים אפקטיביים בגישת מודלי השפה [23,1,42]. אף על פי שמודלים אלו כבר מפעילים מספר גישות עוגן שאילתא, הפעלת השיטות שלנו מביאה לשיפורים נוספים.