Query Anchoring Using Discriminative Query Models

Saar Kuzi
saarkuzi@campus.technion.ac.il

Anna Shtok
annabel@tx.technion.ac.il

Oren Kurland
kurland@ie.technion.ac.il

Technion — Israel Institute of Technology

ABSTRACT

Pseudo-feedback-based query models are induced from a result list of the documents most highly ranked by initial search performed for the query. Since the result list often contains much non-relevant information, query models are anchored to the query using various techniques. 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 generative query models can improve retrieval performance.

1. INTRODUCTION

There is a large body of work on devising pseudo-feedback-based query models (e.g., expanded query forms) [5]. The models are created using information induced from a result list of the documents most highly ranked by some initial search performed in response to the query. The goal is to create a more effective representation of the presumed information need than the query which is often very short. A case in point, query models can help to bridge the vocabulary mismatch between the query and relevant 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 [11, 13, 20]. Thus, a query model induced from these documents can drift away from the information need [22]. 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., [23, 4, 1, 33, 21, 6]). Using the original query model as a prior for the pseudo-feedback-based query model is another example of direct anchoring [27, 28].

ICTIR '16, September 12-16, 2016, Newark, DE, USA

DOI: http://dx.doi.org/10.1145/2970398.2970402

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., [2, 30, 33, 1, 31]). 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 [17, 1, 27, 25]. 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.

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.

More specifically, the proposed query model is produced by training a pairwise learning-to-rank method using the initial result list ranking. The resultant model is composed of terms whose presence in documents is either positively or negatively correlated with the initial result list ranking. Accordingly, these terms can be used for query anchoring.

We demonstrate the merits of applying several methods, which use our new query model, on two highly effective generative query models: the relevance model [17, 1] and the mixture model [32]. Although these models employ several query anchoring approaches, applying our methods results in performance improvements.

2. RELATED WORK

As already noted, a few query anchoring approaches have been proposed in past work. We show that using our methods in addition to several of these most commonly used and effective approaches — interpolation with the original query model [23, 4, 1, 33, 21][1], term clipping [2, 30, 33, 1, 31], and differential weighting of documents in the result list [17, 1, 27, 25] — helps to further improve retrieval performance.

Fusing the initial ranking with the ranking produced by using the pseudo-feedback-based query model was suggested

[1]There has been work on tuning the interpolation on a per query basis when constructing the query model from *true*, rather than pseudo, relevant documents [21].

for indirect query anchoring [34]. 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). Our methods are shown to substantially outperform this fusion approach [34].

There are various methods for improving the quality of the pseudo feedback result list used for inducing a query model (e.g., [22, 24, 18, 12, 15]). The discriminative query model used by our methods can be applied to any ranked list. Thus, our methods are complementary to these past approaches, and more generally, to methods that improve an initial pseudo-feedback-based query model (e.g., [3, 8]).

It was shown that among the terms attributed the highest importance by a pseudo-feedback-based query model, some are highly effective for retrieval while others can be detrimental [3, 29]. Accordingly, a supervised term classification approach was applied using various term features, the most effective of which were based on proximity to query terms in documents in the result list [3]. In contrast, our approach is unsupervised and applies a learning-to-rank method on documents in the result list. Furthermore, we focus on unigram query models and leave the treatment of term-proximity-based query models for future work.

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 [21, 7]. Our findings provide 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.

3. QUERY MODELS

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.

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 [2, 30, 4, 17, 32, 21, 5]. The goal is to create a query model that represents the underlying information need more effectively than a model based only on the terms in q which is often very short.

As noted above, 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 [11, 13, 20].

We present a novel query anchoring approach. The approach utilizes a newly proposed discriminative query model induced from the result list, $\mathcal{D}_{\text{init}}$. The model constitutes a term-based representation of $\mathcal{D}_{\text{init}}$'s *initial ranking*. Thus, the model can be used to query anchor existing pseudo-feedback-based query models as shown below.

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 [21]. We therefore start by describing the language model notation that will be used throughout this paper. In Section 3.1 we survey two highly effective methods of inducing generative query models. In doing so, we will refer to the query anchoring techniques that these methods employ. Then, in Section 3.2 we describe our novel discriminative query model, and in Section 3.3 present a few methods of applying it on an existing language-model-based query model so as to improve retrieval performance.

Language model notation. We use unigram language models. $p_{MLE}(t|x) \stackrel{def}{=} \frac{\text{tf}(t \in x)}{|x|}$ is the maximum likelihood estimate (MLE) of term t with respect to the text (or text collection) x; $\text{tf}(t \in x)$ is the number of occurrences of t in x; $|x| \stackrel{def}{=} \sum_{t' \in x} \text{tf}(t' \in x)$ is x's length. The MLE can be smoothed, for example, using a Dirichlet prior: $p_{Dir}(t|x) \stackrel{def}{=} \frac{\text{tf}(t \in x) + \mu p_{MLE}(t|D)}{|x| + \mu}$; μ is a free parameter [33]. We compare two language models, θ_1 and θ_2, using cross entropy [16]:

$$CE\left(p(\cdot|\theta_1) \,\middle|\middle|\, p(\cdot|\theta_2)\right) \stackrel{def}{=} -\sum_t p(t|\theta_1) \log p(t|\theta_2); \quad (1)$$

lower values correspond to increased similarity.

3.1 Generative query models

3.1.1 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 [17]. Assuming that $\mathcal{D}_{\text{init}}$ was retrieved using the query likelihood approach [26], which ranks document d by $p(q|d) \stackrel{def}{=} \prod_{t \in q} p_{Dir}(t|d)$, the relevance model RM1 is defined as:

$$p(t|RM1) \stackrel{def}{=} \sum_{d \in \mathcal{D}_{\text{init}}} p_{Dir}(t|d) p(d|q); \quad (2)$$

$p(d|q) \stackrel{def}{=} \frac{p(q|d)}{\sum_{d' \in \mathcal{D}_{\text{init}}} p(q|d')}$ 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 Section 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:

$$p(t|RM3) \stackrel{def}{=} \lambda p_{MLE}(t|q) + (1 - \lambda) p(t|RM1^{clipped}). \quad (3)$$

3.1.2 Mixture model

The mixture model [32] 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 $\mathcal{D}_{\text{init}}$,

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

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:

$$p(t|MM) \stackrel{def}{=} \lambda p_{MLE}(t|q) + (1-\lambda)p(t|\theta_T^{clipped}). \quad (4)$$

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 1) between the model and d's language model is used.

3.2 A discriminative query model

Term clipping (applied by RM3 and MM) and differential impact of documents in $\mathcal{D}_{\text{init}}$ (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 [19], namely SVMrank [14], 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 3.3.

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 [19]. 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., click-through information) [19].

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[3]. We apply SVMrank to find a weight vector $\vec{w}$ defined over V that is the solution for:

$$\begin{aligned} &minimize && \frac{1}{2}\vec{w}\cdot\vec{w} + C\sum_{i,j}\xi_{i,j} \\ &subject\ to: && \\ &\forall i \forall j.\ r(d_i) < r(d_j): && \vec{w}\Phi(d_i) \geq \vec{w}\Phi(d_j) + 1 - \xi_{i,j} \\ &\forall i \forall j.\ r(d_i) < r(d_j): && \xi_{i,j} \geq 0 \end{aligned} \quad (5)$$

Equation 5 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.[4]

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 5 where k is the number of documents in $\mathcal{D}_{\text{init}}$; these constraints correspond to all[5] pairs of documents d_i and d_j in $\mathcal{D}_{\text{init}}$ where d_i is ranked *higher* than d_j (i.e., $r(d_i) < r(d_j)$). 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 5.

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 5 amounts, in spirit[6], to:

$$\begin{aligned} &-CE(p(\cdot|\theta_{\vec{w}_+}) \mid\mid p_{Dir}(\cdot|d_i)) + CE(p(\cdot|\theta_{\vec{w}_-}) \mid\mid p_{Dir}(\cdot|d_i)) \geq \\ &-CE(p(\cdot|\theta_{\vec{w}_+}) \mid\mid p_{Dir}(\cdot|d_j)) + CE(p(\cdot|\theta_{\vec{w}_-}) \mid\mid p_{Dir}(\cdot|d_j)) + \\ &1 - \xi_{i,j}; \end{aligned} \quad (6)$$

[2] A regularized mixture model [28] uses the original query model, $p_{MLE}(t|q)$, 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 [21].

[3] Using $\log p_{Dir}(t_i|d)$ results in cross entropy semantics for the constraints presented in Equation 5.

[4] An alternative hard-margin SVM formulation is not guaranteed to have a solution.

[5] For pairs of documents with exactly the same initial retrieval score in $\mathcal{D}_{\text{init}}$ we do not use a constraint.

[6] We write "in spirit" as the weight vectors $\vec{w}_+$ and $\vec{w}_-$ that are parts of the solution to Equation 5 have to be normalized so as to yield valid probability distributions. Therefore, the values compared in the constraints in Equation 5 are not valid CE values. Yet, the observations made with respect to Equation 6, and consequently to Equation 5, still hold. We use the CE expressions to simplify the discussion.

$\theta_{\vec{x}}$ 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 (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 the generative query models described in Section 3.1. 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 4.3. As a result, the generative query models and the discriminative model are of complementary nature. We leverage this fact in Section 3.3 by designing methods that use the discriminative query model to query anchor the generative query models.

3.3 Using the discriminative query model

Let θ be a generative query model. 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:

$$p(t|\theta_{AnchorPos}) \stackrel{def}{=} \lambda_1 p_{MLE}(t|q) + (1-\lambda_1) p(t|\vartheta_+). \quad (7)$$

If θ is RM1 or θ_T, described in Section 3.1, we will refer to AnchorPos as operating on RM3 and MM, respectively. The reason is that for $\lambda_3 = 0$, Equation 7 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 $\theta_{\vec{w}_+}$.

Our next order of business is devising a method that 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:

$$p(t|\theta_{ClipNeg}) \stackrel{def}{=} \lambda p_{MLE}(t|q) + (1-\lambda) p(t|\vartheta_-). \quad (8)$$

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 8 becomes RM3 and MM, respectively.

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:

$$p(t|\theta_{AnchorClip}) \stackrel{def}{=} \lambda_1 p_{MLE}(t|q) + (1-\lambda_1) p(t|\vartheta). \quad (9)$$

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 1).

4. EVALUATION

We present an evaluation of the methods from Section 3.3 that use the discriminative query model. The methods are applied to the relevance model, RM3 [1], and to the mixture model [32], MM, described in Section 3.1. These two generative query models were the most effective in a study of unigram language-model-based query models [21].

4.1 Experimental setup

The datasets specified in Table 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).

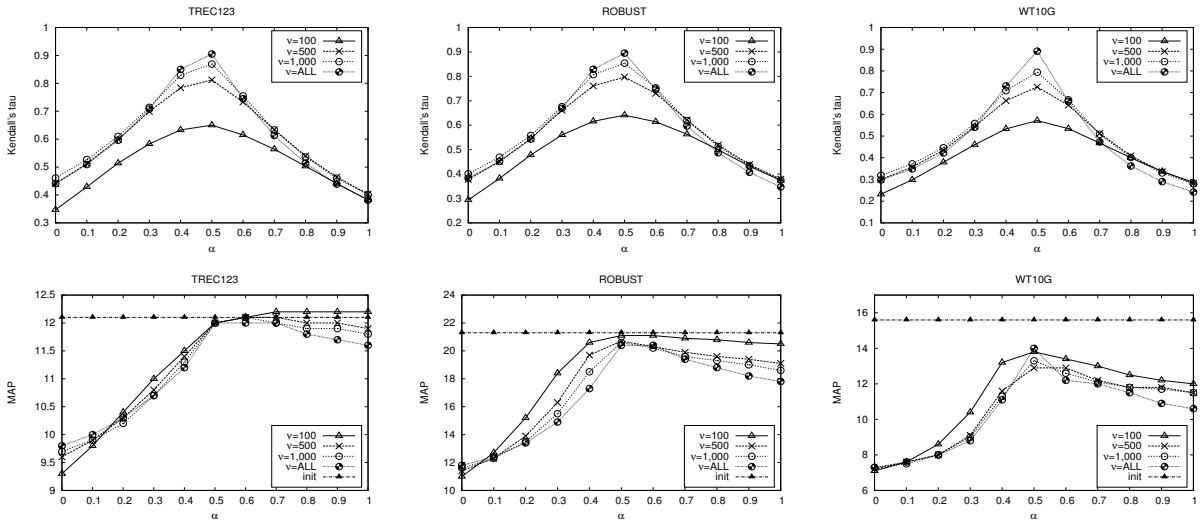

Figure 1: Using the discriminative query model to re-rank an initial list of 100 documents from which it is induced. The Kendall-τ between the initial ranking (init) and the re-ranking, and their MAP(@100), are reported. The positive and negative anchor models are clipped to use ν terms; ν =ALL means no clipping. Note: figures are not to the same scale.

Table 1: TREC datasets used for experiments.

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

The initial result list $\mathcal{D}_{\text{init}}$, which serves for pseudo feedback, is retrieved using a standard language model method [16] which uses cross entropy (see Equation 1): 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 [26] used in the relevance model. Here and after, the Dirichlet smoothing parameter, μ, is set to 1000 [33].

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) [24] 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 4.2.3 we extend the robustness analysis by using risk-reward graphs [8].

As mentioned in Section 3.2, we use SVMrank [14] 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[7]. 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.

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 [34]. Specifically, the top-1000 documents in the initial ranking are fused, using CombMNZ [10], with the top-1000 documents in a ranking produced by using the generative query model [34][8]. 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 [34].

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[9]. 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 [9] 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

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

[8] CombMNZ was more effective in our setting than the alternative interpolation-based fusion method [34].

[9] 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.

Table 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.

	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

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[10]. We show in Section 4.2.3 that clipping a low percentage of negative anchor terms yields worse performance than clipping a high percentage. The parameters λ, λ_1, λ_2 and λ_3 are set to values in $\{0, 0.2, \ldots, 1\}$.

4.2 Experimental results

4.2.1 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 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 1, is applied to $\mathcal{D}_{\text{init}}$ of $k = 100$ documents. We also report MAP(@100) for the rankings[11].

We see in Figure 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 1 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 1 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.

4.2.2 Main result

Table 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 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 × 2 evaluation measures × 2 generative models); the majority of MAP improvements for AnchorPos are statistically significant[12]. 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 2 that using positive anchor terms (AnchorPos) is almost always more effective in terms of retrieval effectiveness (MAP and p@5) than using negative anchor terms (ClipNeg). More gener-

[10]Since $e \neq 0$ for AnchorClip, we enforce clipping.

[11]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.

[12]The relative p@5 performance patterns reflect those for p@20 which was the focus of some work on query expansion [8]. For example, RM3 attained p@20 of 51.8, 36.8 and 26.4 over TREC123, ROBUST and WT10G, respectively. Applying AnchorPos to RM3 yields 52.7, 38.0, and 26.3, respectively, and applying ClipNeg yields 52.4, 37.8, and 25.5, respectively. MM attained p@20 of 50.3, 36.8 and 25.8 for the three corpora; applying AnchorPos on MM yields 51.5, 38.2, and 26.3; applying ClipNeg yields 50.8, 37.0 and 26.3. For ROBUST, the improvements of ClipNeg over RM3 and of AnchorPos over MM are statistically significant.

Table 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.

	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

ally, in most cases, AnchorPos is the most effective method in Table 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 [34]. Yet, the RI of Fusion is in most cases inferior to that of AnchorPos.

Table 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 3.3, integrates the ClipNeg and AnchorPos methods by boosting the probabilities of positive anchor terms as in AnchorPos and clipping negative anchor terms as in ClipNeg. Table 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.

4.2.3 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 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.

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.

Table 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.

	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
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

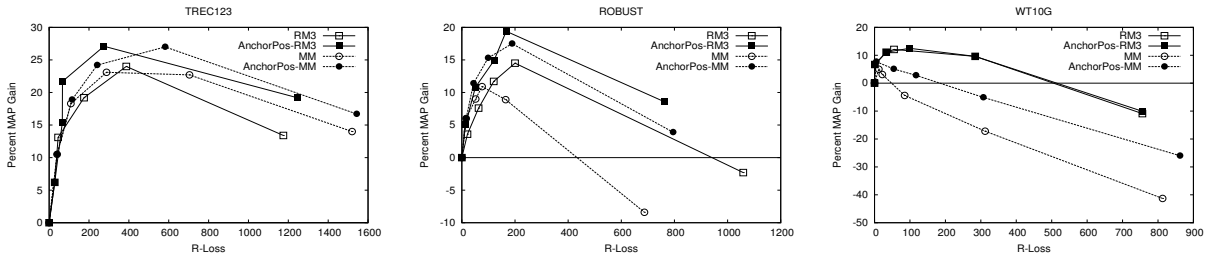

Figure 2: MAP risk-reward curves [8]. Note: figures are not to the same scale.

To further study the performance robustness of AnchorPos, in Figure 2 we present its MAP risk-reward curves [8] 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[13]; 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 2 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.

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 3 contrasts the performance of ClipNeg with that of **ClipRand**;[14] 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 3 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

[13]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.

[14]In contrast to the evaluation results presented in Tables 2, 3 and 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.

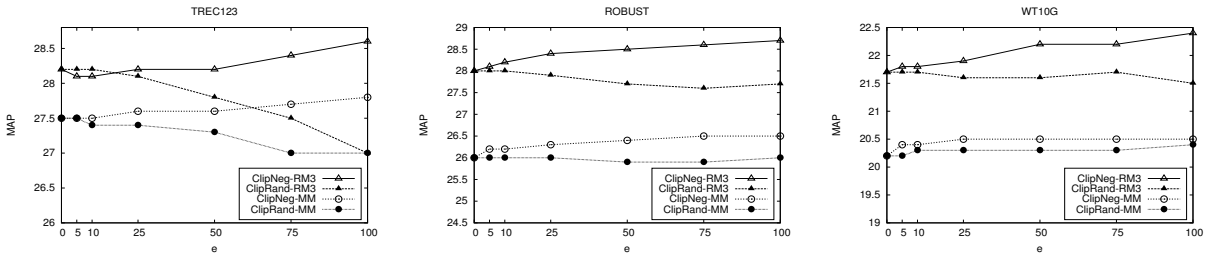

Figure 3: The MAP performance of ClipNeg and ClipRand as a function of the percentage of clipped negative anchor terms (e). ClipRand clips randomly selected terms. ClipNeg and ClipRand are applied on RM3 and MM. Initial list, $\mathcal{D}_{\text{init}}$, of 100 documents is used. Note: figures are not to the same scale.

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.

4.3 The discriminative vs. the generative query models

To illustrate the differences between the discriminative and generative query models, we provide in Figure 4 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 3, 4 and 7). The models resulting from the interpolation are RM3 (based on RM1), MM (based on θ_T) and Q+Pos (based on $\theta_{\vec{w}_+}$; see Section 4.2.3 for details) which interpolates the positive anchor model ($\theta_{\vec{w}_+}$) with the original query model.

Figure 4 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 $\mathcal{D}_{\text{init}}$. 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 4. 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 4 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. 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 3.3.

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 4 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 [21, 5, 7].)

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 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. 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 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 3.3 and whose effectiveness was demonstrated above.

5. CONCLUSIONS AND FUTURE WORK

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. The resultant query model constitutes a term-based representation of the list ranking.

Figure 4: 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.

We demonstrated the empirical merits of methods using the discriminative model to query anchor existing query models: emphasizing (clipping) terms that are positively (negatively) correlated with the initial ranking.

We intend to apply the discriminative query model on additional pseudo-feedback-based query models, and study the utilization of additional learning-to-rank methods to induce a term-based representation from a retrieved list.

Acknowledgments. We thank the reviewers for their comments. This work was supported in part by the Israel Science Foundation (grant no. 433/12), the Technion-Microsoft Electronic Commerce Research Center and by the Irwin and Joan Jacobs Fellowship for graduate students.

6. REFERENCES

[1] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. UMASS at TREC 2004 — novelty and hard. In *Proc. of TREC-13*, 2004.

[2] C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic query expansion using SMART: TREC3. In *Proc. of TREC-3*, pages 69–80, 1994.

[3] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proc. of SIGIR*, pages 243–250, 2008.

[4] C. Carpineto, R. de Mori, G. Romano, and B. Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, 19(1):1–27, 2001.

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

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

[7] S. Clinchant and É. Gaussier. A theoretical analysis of pseudo-relevance feedback models. In *Proc. of ICTIR*, 2013.

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

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

[10] E. A. Fox and J. A. Shaw. Combination of multiple searches. In *Proc. of TREC-2*, 1994.

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

[12] B. He and I. Ounis. Finding good feedback documents. In *Proc. of CIKM*, pages 2011–2014, 2009.

[13] B. He and I. Ounis. Studying query expansion effectiveness. In *Proc. of ECIR*, pages 611–619, 2009.

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

[15] M. Keikha, J. Seo, W. B. Croft, and F. Crestani. Predicting document effectiveness in pseudo relevance feedback. In *Proc. of CIKM*, pages 2061–2064, 2011.

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

[17] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proc. of SIGIR*, pages 120–127, 2001.

[18] K.-S. Lee, W. B. Croft, and J. Allan. A cluster-based resampling method for pseudo-relevance feedback. In *Proc. of SIGIR*, pages 235–242, 2008.

[19] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3), 2009.

[20] Y. Lv and C. Zhai. Adaptive relevance feedback in information retrieval. In *Proc. of CIKM*, pages 255–264, 2009.

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

[22] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *Proc. of SIGIR*, pages 206–214, 1998.

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

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

[25] J. Seo and W. B. Croft. Geometric representations for multiple documents. In *Proc. of SIGIR*, pages 251–258, 2010.

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

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

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

[29] R. Udupa, A. Bhole, and P. 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.

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

[31] Z. Ye, B. He, X. Huang, and H. Lin. Revisiting Rocchio's relevance feedback algorithm for probabilistic models. In *Proc. of AIRS*, pages 151–161, 2010.

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

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

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