Inf Retrieval (2011) 14:593–616
DOI 10.1007/s10791-011-9168-6

A study of the integration of passage-, document-, and cluster-based information for re-ranking search results

Eyal Krikon · Oren Kurland

Received: 17 December 2010 / Accepted: 28 April 2011 / Published online: 20 May 2011

Abstract Cluster-based and passage-based document retrieval paradigms were shown to be effective. While the former are based on utilizing query-related corpus context manifested in clusters of similar documents, the latter address the fact that a document can be relevant even if only a very small part of it contains query-pertaining information. Hence, cluster-based approaches could be viewed as based on "expanding" the document representation, while passage-based approaches can be thought of as utilizing a "contracted" document representation. We present a study of the relative benefits of using each of these two approaches, and of the potential merits of their integration. To that end, we devise two methods that integrate whole-document-based, cluster-based and passage-based information. The methods are applied for the *re-ranking* task, that is, re-ordering documents in an initially retrieved list so as to improve precision at the very top ranks. Extensive empirical evaluation attests to the potential merits of integrating these information types. Specifically, the resultant performance substantially transcends that of the initial ranking; and, is often better than that of a state-of-the-art pseudo-feedback-based query expansion approach.

Keywords Ad hoc retrieval · Re-ranking · Clusters · Cluster-based language models · Passages · Passage-based language models

1 Introduction

A standard paradigm to addressing the ad hoc (query-based) retrieval task is devising document and query representations, and using their similarity to induce ranking. In the vector space model, for example, a vector representing the query and that representing a

E. Krikon · O. Kurland (✉)
Faculty of Industrial Engineering and Management, Technion,
Israel Institute of Technology, 32000 Haifa, Israel
e-mail: kurland@ie.technion.ac.il

E. Krikon
e-mail: krikon@tx.technion.ac.il

document can be compared using the cosine similarity measure (Salton et al. 1975). In the language modeling framework, the KL divergence between the query and document language models often serves for ranking (Lafferty and Zhai 2001).

There has been much work on devising query representations, document representations, and similarity measures. For instance, various approaches for automatic query expansion have been developed (Buckley et al. 1994; Xu and Croft 1996; Lavrenko and Croft 2001; Zhai and Lafferty 2001a). Furthermore, there is a large body of work on integrating representations and similarity measures (Croft 2000b). Our focus in this paper is on the document side, that is, (specific) document representations and their integration.

The document representation task has attracted quite a lot of research attention throughout the history of information retrieval. The effectiveness of using manually versus automatically selected terms to index the document with was studied (Salton and Lesk 1968). Using specific index terms versus using the entire document, or its abstract or title, was also explored (Fisher and Elchesen 1972; McGill et al. 1979; Katzer et al. 1982). In many cases, the conclusion was that integrating different document representations can yield retrieval performance that is better than that of using each representation alone (Katzer et al. 1982). Cognition-based arguments, for example, were proposed to support the merits of such integration (Ingwersen 1994). Another form of document representation that was explored is based on automatic *summarization*, performed in a query-independent (Radev et al. 2002) or query-dependent (Tombros and Sanderson 1998) manner. Such representations can help the user, for example, to effectively examine search results.

Document representation can also be based on information that is not part of the document itself, e.g., so as to cope with the vocabulary mismatch between relevant documents and the query. For example, a document can be "expanded" using bibliographic information (Salton 1963; Kwok 1975), or a thesaurus (Joyce and Needham 1958). Alternatively, an expanded document form can be derived using similar documents in the corpus (Singhal and Pereira 1999; Kurland and Lee 2004; Liu and Croft 2004; Tao et al. 2006), or by utilizing topic-based information that is induced from the corpus (Deerwester et al. 1990; Hofmann 1999; Wei and Croft 2006; Yi and Allan 2009). On the Web, hyperlink (and hypertext) information can be used to enrich the document representation (McBryan 1994; Craswell et al. 1999; Kraaij et al. 2002; Ogilvie and Callan 2003; Metzler et al. 2009). Recently, temporal versions of the document have been used to form a representation (Elsas and Dumais 2010).

In that respect, the work on *cluster-based retrieval* could be viewed as representing a conceptual approach that treats a document as part of its corpus context, rather than in isolation. Examples include enriching a document model using information induced from clusters of similar documents (Singhal and Pereira 1999; Kurland and Lee 2004; Liu and Croft 2004, 2006b; Tao et al. 2006); and more generally, using document-cluster associations to identify documents pertaining to the query (Jardine and van Rijsbergen 1971; Croft 1980; Voorhees 1985; Willett 1985; Kurland and Lee 2004; Liu and Croft 2004, 2006b, 2008; Kurland and Lee 2006; Yang et al. 2006).

A conceptually opposite approach to expanding a document representation is manifested in *passage-based* document ranking models. The goal of such methods is to address the fact that a long and/or topically heterogeneous relevant document might contain only a small part (passage) with information pertaining to the query. A common retrieval method is ranking a document by the highest query-similarity exhibited by any of its passages (Salton et al. 1993; Callan 1994; Wilkinson 1994; Liu and Croft 2002).

Thus, the cluster-based and passage-based document ranking paradigms could be viewed as two extremes of the spectrum of approaches utilizing different document

representations, that is, expansion versus contraction. Furthermore, these paradigms essentially address different, yet potentially complementary, goals: exploiting corpus context versus handling long/heterogeneous documents. Naturally, then, the following research questions rise. Can cluster-based and passage-based information be effectively integrated, along with whole-document-based information, so as to improve upon using each alone? are there cases wherein using cluster-based information is clearly more effective than using passage-based information and vice versa? We note that although demonstrated to be effective for document retrieval, cluster-based and passage-based information have been utilized separately in different retrieval methods.

To address the research questions just stated, we perform the following study. We devise two retrieval methods that integrate whole-document-based, cluster-based, and passage-based information. The first is a language-model-based (LM) method that integrates language models induced from documents, clusters, and passages. The method generalizes some previously proposed ranking methods that utilize either passage-based or cluster-based information, but not both. As such, the LM method enables us to thoroughly study the relative performance contributions of each of the information types it leverages. The second method that we present is based on a discriminative approach. Specifically, we use a *learning-to-rank* algorithm (Joachims 2002; Liu 2009) that utilizes information induced from documents, their passages, and clusters.

We use the proposed methods for the *re-ranking* task, which has attracted quite a lot of research attention lately (Liu and Croft 2004, 2006a, b, 2008; Diaz 2005; Kurland and Lee 2005, 2006; Yang et al. 2006; Kurland 2009). That is, re-ordering documents in an initially retrieved list so as to improve precision at the very top ranks. Extensive empirical evaluation performed using six TREC corpora shows that both the LM method and the learning-to-(re-)rank approach are highly effective in re-ranking. Specifically, the performance transcends that of a state-of-the-art cluster-based re-ranking method, and, that of a commonly used passage-based document ranking approach. Furthermore, the performance is often better than that of a state-of-the-art pseudo-feedback-based query expansion approach. The latter comparison could conceptually be viewed as contrasting two paradigms: enriching query representation versus enriching (and/or contracting) document representation.

The findings with respect to the questions posted above that emerge in the study we performed are as follows. Using passage-based information is much more effective than using cluster-based information for corpora containing very long and topically-heterogeneous documents; e.g., TREC's FR corpus. Yet, even for such corpora, integration of the two types of information can yield performance that substantially transcends that of using each alone. For the rest of the corpora we examine, using cluster-based information is much more effective than using passage-based information, but yet again, their integration can yield improved performance. More generally, we show that integration of whole-document, cluster and passage-based information can yield clear merits over using any subset of these three information types. Finally, we show that while simple learning, performed across queries, of the relative impact of these information types yields highly effective re-ranking performance, there is large room for improvement that can potentially be attained by devising methods for setting this balance on a per-query basis.

All in all, we note that our contributions are two fold. First, we study the relative merits of using whole-document-, cluster-, and passage-based information, and their integration, in the ad hoc retrieval setting. Second, we present re-ranking methods that integrate these three information types and which yield high precision at top ranks. Naturally, users would like to see the documents pertaining to their information needs at the highest ranks of the

retrieved lists. Furthermore, for applications such as question answering that rely on search as an intermediate step high precision at top ranks is important (Voorhees 2002). Finally, we note that while the focus of this paper is on the "document side", further future performance improvements can potentially be attained by integration with techniques relying on the "query side"; e.g., query expansion.

2 Related work

There is a large body of work on re-ranking an initially retrieved list using information induced from clusters of documents in the list (Willett 1985; Liu and Croft 2004, 2006a, b, 2008; Kurland and Lee 2006; Yang et al. 2006; Kurland 2009). As will be shown in Sect. 3.1.1, our main (re-)ranking method generalizes a state-of-the-art cluster-based re-ranking model (Kurland 2009), which does not utilize passages. The relative performance merits of our model, which utilizes passage-based information on top of cluster-based information, are demonstrated in Sect. 4.3.

Graph-based re-ranking methods utilizing inter-item similarities have become quite common (e.g., Baliński and Daniłowicz 2005; Diaz 2005; Kurland and Lee 2005; Kurland and Lee 2006; Yang et al. 2006; Krikon et al. 2009). Specifically, document centrality (Kurland and Lee 2005), cluster centrality (Kurland and Lee 2006), and passage centrality (Krikon et al. 2009) induced over such graphs, were shown to be effective for re-ranking documents. Extending our model by using such centrality measures is a future venue we aim to explore. Indeed, the merits of such practice were demonstrated in work that also uses passages as proxies for documents (Krikon et al. 2009); however, cluster-based information, which is highly effective for re-ranking as we show in Sect. 4.3, was not utilized (Krikon et al. 2009). Re-ranking an initially retrieved list using inter-document similarities was also employed for searching over digital libraries (Van and Beigbeder 2008), cross-lingual retrieval (Diaz 2008), and fusion of retrieved lists (Meister et al. 2010).

A common passage-based document retrieval method is ranking a document by the highest query-similarity that any of its passages exhibits (Callan 1994; Wilkinson 1994; Kaszkiel and Zobel 2001; Liu and Croft 2002; Bendersky and Kurland 2008; Na et al. 2008); and, interpolating this similarity score with a document-query similarity score (Buckley et al. 1994; Callan 1994; Wilkinson 1994; Cai et al. 2004; Bendersky and Kurland 2008). Our re-ranking model generalizes these methods as will be shown in Sect. 3.1.1. Furthermore, we show in Sect. 4.3 that the model posts much better performance than that of these methods.

There is some work on discriminative models for passage-based document retrieval (Wang and Si 2008). In contrast to the learning-to-re-rank approach that we present, cluster-based information, which is highly effective for re-ranking as we show in Sect. 4.3, is not utilized.

Utilizing information induced from passages could be viewed as a means for exploiting relationships between terms that are somewhat close to each other in the text. Using Markov random fields (Metzler and Croft 2005), positional language models (Lv and Zhai 2009; Zhao and Yun 2009), and approaches that utilize the document structure (e.g., for XML documents) (Beigbeder et al. 2009) has also been suggested for exploiting information induced from inter-term and term-(document) position proximities. In Sect. 4.3 we use unigram language models in our re-ranking approaches so as to facilitate the comparison with previous work in the language modeling framework on (i) passage-based

(Liu and Croft 2002), (ii) cluster-based (Kurland and Lee 2004; Kurland 2009), and (iii) relevance-model-based (Lavrenko and Croft 2001; Abdul-Jaleel et al. 2004) retrieval that used these unigram models. However, we hasten to point out that the Markov random field approach, and/or positional language models, can be used in our methods for estimating the document-query and passage-query "match" so as to potentially improve performance—a venue which we leave for future work.

Previous work on passage-based retrieval has focused on identifying and utilizing different types of passages. For example, (i) *discourse passages* are inferred from document markup (e.g., sentences or SGML tags) (Salton and Buckley 1991; Callan 1994; Wilkinson 1994; Cai et al. 2004; Hussain 2004), (ii) *semantic passages* are induced based on presumed topic shifts in a document (Hearst and Plaunt 1993; Mittendorf and Schäuble 1994; Ponte and Croft 1997; Denoyer et al. 2001; Jiang and Zhai 2004), and (iii) fixed (or variable) length passages are simply *windows* of consecutive terms in the document (Callan 1994; Kaszkiel and Zobel 1997; Liu and Croft 2002; Wade and Allan 2005; Na et al. 2008; Wang and Si 2008). While we focus on the latter in the evaluation presented in Sect. 4, as those were shown to be highly effective for document retrieval (see Sect. 4.2 for further details), we note that our re-ranking methods are not committed to any specific type of passages.

Furthermore, there is a large body of work on devising passage-based (Liu and Croft 2002; Abdul-Jaleel et al. 2004; Murdock and Croft 2005; Wade and Allan 2005; Bendersky and Kurland 2008) and cluster-based (Liu and Croft 2006b, 2008; Tao et al. 2006) language models. These language models could be used by our models, which are not committed to a specific language-model induction technique, so as to potentially improve their performance.

3 Re-ranking search results

Notational conventions We use q, d, and $\mathcal{D}$ to denote a query, a document, and a corpus of documents, respectively. Our goal is to re-rank an initial list, $\mathcal{D}_{\text{init}}$ ($\subset \mathcal{D}$), which was retrieved by some search algorithm in response to q, so as to improve precision at top ranks. To that end, a set of *clusters* of similar documents, $Cl(\mathcal{D}_{\text{init}})$, created from documents in $\mathcal{D}_{\text{init}}$ by some clustering algorithm, is utilized; c is used to denote a cluster. Our re-ranking methods also exploit information induced from *passages* in documents. We use g to denote a passage, and write $g \in d$ if g is part of d. The methods we present are not committed to a specific clustering algorithm, nor to a specific technique of segmenting documents to passages.

3.1 A language-model-based approach

We rank the documents in $\mathcal{D}_{\text{init}}$ using a probabilistic approach. Specifically, we aim to estimate $p(d|q)$—the probability that d is relevant to the information need expressed by q. Assuming a uniform prior distribution over documents, the following rank equivalence holds

$$p(d|q) \stackrel{\text{rank}}{=} p(q|d). \tag{1}$$

In the language-modeling framework (Ponte and Croft 1998; Croft and Lafferty 2003), for example, $p(q|d)$ is regarded as the probability of generating the terms in q by a language

model induced from d. However, we hasten to point out that the derivation to follow is not committed to any specific paradigm of estimating probabilities, albeit we will use language-model-based estimates for implementation.[1]

Clusters in $Cl(\mathcal{D}_{\text{init}})$ could potentially be thought of as representing query-related "aspects", by the virtue of the way they are created, that is, from documents retrieved in response to the query (Liu and Croft 2004; Kurland and Lee 2006). We therefore use clusters as proxies for d (Kurland and Lee 2004):

$$p(q|d) = \sum_{c \in Cl(\mathcal{D}_{\text{init}})} p(q|d,c)p(c|d). \tag{2}$$

To estimate $p(q|d,c)$, we use a simple mixture governed by a free parameter λ_{clust}: $(1-\lambda_{clust})p(q|d) + \lambda_{clust}p(q|c)$. As $p(c|d)$ is a probability distribution over $Cl(\mathcal{D}_{\text{init}})$, the universe of clusters that we consider, we can use some probability algebra to derive a previously-proposed cluster-based retrieval algorithm (Kurland and Lee 2004; Kurland 2009):[2]

$$Score_{clust}(d|q) \stackrel{\text{def}}{=} (1-\lambda_{clust})p(q|d) + \lambda_{clust} \sum_{c \in Cl(\mathcal{D}_{\text{init}})} p(q|c)p(c|d). \tag{3}$$

Consequently, document d is highly ranked if it exhibits a good "match" to the query, as measured by $p(q|d)$, and if it is strongly associated with clusters of documents in $\mathcal{D}_{\text{init}}$ (as measured by $p(c|d)$) that are a good "match" to the query ($p(q|c)$).

A potential shortcoming of the ranking function in (3) is that d is treated as a whole unit. Indeed, it could be the case that only a small part (passage) of d contains information pertaining to q, and d is still deemed relevant—e.g., by TREC's relevance-judgment regime (Voorhees and Harman 2005). More generally, since passages could be considered as more coherent units than documents, they can potentially serve as proxies in estimating the document-query match—$p(q|d)$ in our case. For example, some previous work (Bendersky and Kurland 2008) has demonstrated the merits in using

$$Score_{psg}(d|q) \stackrel{\text{def}}{=} (1-\lambda_{psg})p(q|d) + \lambda_{psg} \max_{g_i \in d} p(q|g_i) \tag{4}$$

as an estimate for $p(q|d)$; λ_{psg} is a free parameter. Such an approach can help to address the above-mentioned scenario of having a single passage in a document that contains query-pertaining information.

To integrate information induced from *both* passages and clusters, we can use the estimate from (4) for $p(q|d)$ in (3) so as to get:

$$\begin{aligned} Score(d|q) \stackrel{\text{def}}{=} & (1-\lambda_{clust})(1-\lambda_{psg})p(q|d) \\ & + (1-\lambda_{clust})\lambda_{psg} \max_{g_i \in d} p(q|g_i) + \lambda_{clust} \sum_{c \in Cl(\mathcal{D}_{\text{init}})} p(q|c)p(c|d). \end{aligned} \tag{5}$$

Algorithm The probabilities in (5) can be estimated in various ways. Here, we follow common practice in the language-modeling framework (Ponte and Croft 1998; Croft and Lafferty 2003). Specifically, we use a language-model-based estimate, $p_y(x)$, for $p(x|y)$;

[1] We do not assume an underlying *generative* theory in contrast to Lavrenko and Croft (2001) and Lavrenko (2004), inter alia.

[2] The shift in terminology from "probability" to "score" is intended to emphasize the transition from model-based probabilities to estimates of such probabilities.

$p_y(x)$ is based on the probability of generating the text x by a language model induced from text y. (Specific language-model induction details are described in Sect. 4.1). Thus, we arrive to our cluster-document-passage language-model-based re-ranking algorithm, henceforth referred to as **CDPlm**:

$$\begin{aligned} Score_{CDPlm}(d|q) \stackrel{\text{def}}{=} & (1-\lambda_{clust})(1-\lambda_{psg})p_d(q) \\ & + (1-\lambda_{clust})\lambda_{psg} \max_{g_i \in d} p_{g_i}(q) + \lambda_{clust} \sum_{c \in Cl(\mathcal{D}_{\text{init}})} p_c(q)p_d(c). \end{aligned} \quad (6)$$

CDPlm is a three-component mixture model. The first component is based on the direct "match" between d and q. The second component uses d's passage that exhibits the best "match" to q as a proxy in estimating d's "match" to q. The third component uses clusters as proxies for d.

3.1.1 Generalizing previous models

The CDPlm method, and more generally, the ranking criterion in (5) on which it is based, generalize various previously proposed document ranking methods. For example, setting $\lambda_{clust} = \lambda_{psg} = 0$ in (6)—i.e., using no passage-based and cluster-based information—yields the standard language model approach (Ponte and Croft 1998). Alternatively, setting only $\lambda_{psg} = 0$, hence using no passage-based information, we get, as mentioned above, a previously-proposed cluster-based ranking model (Kurland and Lee 2004), with which we empirically compare CDPlm in Sect. 4.3.

Setting $\lambda_{clust} = 0$, that is, ignoring cluster-based information, yields a commonly used passage-based document ranking approach (Buckley et al. 1994; Callan 1994; Cai et al. 2004; Wilkinson 1994) with which we empirically compare CDPlm in Sect. 4.3; further setting $\lambda_{psg} = 1$ yields another commonly used passage-based document ranking principle (Callan 1994; Kaszkiel and Zobel 2001; Wilkinson 1994; Liu and Croft 2002; Bendersky and Kurland 2008).

3.2 Learning to re-rank

The CDPlm method is based on estimating the probability of document relevance using language-model estimates. We now turn to devise an alternative re-ranking method that is based on a discriminative approach, but which also uses language-model-based estimates. Specifically, we employ a commonly used *learning to rank* method, SVM^{rank} (Joachims 2006), which uses support vector machines. The learner is presented with examples of queries and rankings of the initial document lists for these queries; the rankings are determined using relevance judgments. The learned ranking function is then used to re-rank an initial list for a new query.

Each document d in the initial list is represented by a vector of features that presumably indicate its relevance to the query. A weight vector for the features is learned so as to discriminate non-relevant documents from relevant documents for (roughly speaking) as many such pairs as possible in the training set (Joachims 2002).[3] We use a linear kernel

[3] Experimental results—specific numbers are omitted as they convey no additional insight—show that SVM^{rank} that optimizes MAP (Yue et al. 2007) yields performance comparable to that of standard SVM^{rank} (Joachims 2006) in our experimental setup.

SVM; hence, the learned function is linear in features. Now, recall that our CDPlm method is a linear mixture of three information types (whole-document-based, passage-based, and cluster-based). Hence, we use these three as features representing a document with respect to a query so as to study whether the balance between them can be learned using a discriminative approach as that employed by SVM^{rank}:

1. The document-based feature:

$$\textbf{DocFeature}(d) \stackrel{\text{def}}{=} p_d(q).$$

2. The cluster-based feature:

$$\textbf{ClustFeature}(d) \stackrel{\text{def}}{=} \sum_{c \in Cl(\mathcal{D}_{\text{init}})} p_c(q) p_d(c).$$

3. The passage-based feature:

$$\textbf{PsgFeature}(d) \stackrel{\text{def}}{=} \max_{g_i \in d} p_{g_i}(q)$$

The resultant (re-)ranking model is denoted **CDPsvm**.

We note that using binary features that indicate whether document d is among the top-ranked documents with respect to a specific feature value, as originally proposed (Joachims 2002), has shown no merit.[4] Furthermore, adding features that utilize some types of passage-based and document-based information other than those utilized by CDPlm has yielded no performance gains. For example, using in addition to the features described above passage centrality and document centrality induced over similarity-based graphs—as those were shown to be effective for re-ranking (Kurland and Lee 2005; Krikon et al. 2009)—has not yielded performance improvements.

4 Evaluation

In what follows we present an evaluation of the performance of the CDPlm and CDPsvm methods. The rest of this section is organized as follows. In Sect. 4.1 we describe the language model estimate used for implementation. Section 4.2 provides details with respect to the experimental setup. Section 4.3 presents the results of our experiments.

4.1 Language-model induction

In this section, we refer to documents, passages, and queries as term sequences. A cluster is represented by the long document that results from concatenating its constituent documents (Kurland and Lee 2004; Liu and Croft 2004). The order of concatenation has no effect since we use unigram language models that assume term independence.

Let $p_z^{Dir[\mu]}(\cdot)$ be the Dirichlet-smoothed unigram language model induced from text z (a query, document, cluster, or passage) with smoothing parameter μ (Zhai and Lafferty 2001b). We use a previously-proposed estimate based on the KL divergence (Lafferty and Zhai 2001; Kurland and Lee 2004, 2005):

[4] Normalizing feature values across documents per query has shown no merit as well.

Table 1 TREC corpora used for experiments

Corpus	# of docs	Avg. doc length	Inter-passage similarity	Queries	Disks
AP	242,918	464	0.113	51–150	1–3
FR	45,820	1,498	0.098	51–150	1–2
SJMN	90,257	414	0.108	51–150	3
TREC8	528,155	481	0.106	401–450	4–5
WSJ	173,252	452	0.104	151–200	1–2
WT10G	1,692,096	611	0.061	451–550	WT10G

$$p_y(x) \stackrel{\text{def}}{=} \exp\left(-D\left(p_x^{Dir[0]}(\cdot)\middle\|p_y^{Dir[\mu]}(\cdot)\right)\right).$$

The estimate was shown to be effective in work on cluster-based retrieval (Kurland and Lee 2004; Kurland 2009) with which we compare our methods, and passage-based retrieval (Krikon et al. 2009). For example, the estimate addresses underflow and length-based issues that result from assigning language-model probabilities to long sequences of text (Lafferty and Zhai 2001; Lavrenko et al. 2002; Kurland and Lee 2005), e.g., $p_d(c)$. While the estimate does not constitute a probability distribution—as is the case for unigram language models—normalizing it to this end yields no performance merits as was the case in some previous work (Krikon et al. 2009; Kurland 2009).

4.2 Experimental setup

We conducted experiments using the TREC corpora specified in Table 1. For each corpus we report the average document length of a document in the corpus, and the average similarity between passages in a document in the initial list, $\mathcal{D}_{\text{init}}$, to be re-ranked (further details below). The latter is computed by $\frac{1}{|\mathcal{D}_{\text{init}}|}\sum_{d\in\mathcal{D}_{\text{init}}}\frac{\sum_{g_i\in d, g_j\in d} p_{g_i}(g_j)}{m(d)^2}$, where $m(d)$ is the number of passages in d and $|\mathcal{D}_{\text{init}}|$ is the number of documents in $\mathcal{D}_{\text{init}}$.[5] The motivation for using these corpora is based on the different types of documents that they contain (news articles, federal register records, and Web pages), the varying average document length and presumed document "homogeneity" (as measured by inter-passages similarities) that can affect the relative effectiveness of document-based, passage-based and cluster-based retrieval; and, compliance with previous work on cluster-based re-ranking (Kurland 2009) and passage-based retrieval (Callan 1994; Liu and Croft 2002) that used some of these corpora and with which we compare our models.

Specifically, AP, SJMN and WSJ are news corpora. TREC8, which is considered a hard benchmark (Voorhees 2005), is mainly composed of news documents, but also contains federal register records. FR is composed of only federal register records. Furthermore, passage-based document ranking methods are known to be more effective than whole-document-based approaches for FR (Callan 1994; Liu and Croft 2002; Bendersky and Kurland 2008; Wang and Si 2008). This finding is often attributed to the fact that the FR documents are very long and "heterogeneous". Indeed, the average document length for

[5] Since $p_x(y)$ is an asymmetric function, we consider both $p_{g_u}(g_v)$ and $p_{g_v}(g_u)$ in the average. The similarity of a passage to itself is also considered as it serves as a regularization factor that can help address the fact, for example, that many documents in the news corpora have a single passage and omitting these (which might be considered "homogeneous") results in somewhat biased statistics.

FR is much higher than that for other corpora; and, the average within-document inter-passage similarity is quite low with respect to that for the news corpora. We come back to these points later on. WT10G is a Web corpus that contains quite long (on average) documents. Furthermore, the Web documents are quite "heterogeneous" as measured by the within-document inter-passage similarities.

We used titles of TREC topics for queries.[6] Tokenization and Porter stemming were applied using the Lemur toolkit (http://www.lemurproject.org). Stop words were not removed. The Lemur and Zettair (http://www.seg.rmit.edu.au/zettair) toolkits were used for experiments.

We use the experimental setup proposed in some previous work on re-ranking (Kurland and Lee 2005, 2006; Kurland 2009; Krikon et al. 2009). The list $\mathcal{D}_{\text{init}}$, upon which re-ranking is performed, is set to the 50 documents in the corpus that yield the highest $p_d(q)$—i.e., a standard language-model-based approach. We note that re-ranking methods that utilize inter-document similarities—in our case, using information induced from document clusters—are known to be most effective when employed over relatively short retrieved lists (Diaz 2005; Kurland 2006). The document language-model smoothing parameter, μ, is set to optimize MAP (at 1000) so as to have an initial list of reasonable quality. In Sect. 4.3 we show that when employed over such a reasonable ranking, our re-ranking methods can yield performance that is better than that of state-of-the-art retrieval methods, whether used to rank the entire corpus or only re-rank the initial list.

The goal of re-ranking methods is to improve precision at the very top ranks. Therefore, we focus on the precision of the top 5 and 10 documents (p@5 and p@10, respectively) as evaluation measures. Statistically significant performance differences are determined using the two-tailed paired t test at a confidence level of 95% (Sanderson and Zobel 2005; Smucker et al. 2007).

As mentioned above, our methods are not committed to a specific type of passages and clusters. We use half-overlapping windows of 150 terms for passages, as these were shown to be effective (e.g., in comparison to other types of passages and in comparison to windows of 50 and 25 terms) in work on passage-based document retrieval (Callan 1994; Wang and Si 2008), specifically, in the language modeling framework (Liu and Croft 2002; Bendersky and Kurland 2008; Krikon et al. 2009).

To cluster $\mathcal{D}_{\text{init}}$, we employ a commonly used nearest-neighbor-based approach that yields overlapping clusters (Griffiths et al. 1986; Kurland and Lee 2006; Liu and Croft 2006a; Kurland 2009). For each d ($\in \mathcal{D}_{\text{init}}$) we define a cluster that contains d and the $k-1$ documents d_i in $\mathcal{D}_{\text{init}}$ ($d_i \neq d$) that yield the highest $p_{d\_i}(d)$. We use clusters of $k = 10$ documents, as such small clusters were shown to be effective in work on cluster-based retrieval, specifically, for the re-ranking task (Kurland and Lee 2006; Liu and Croft 2006a; Kurland 2009).

Parameters The smoothing parameter, μ, is set to 2000 (Zhai and Lafferty 2001b) in all methods, except for estimating $p_d(q)$, where we use the value chosen for creating $\mathcal{D}_{\text{init}}$ so as to maintain consistency with the initial ranking.

The CDPlm method incorporates two free parameters, λ_{clust} and λ_{psg}, which control the relative impact of cluster-based and passage-based information, respectively. To thoroughly study the relative merits of using these information types, and the overall resultant effectiveness of CDPlm, we use the following experimental settings.

[6] Queries with no relevant documents were not considered.

In Sect. 4.3.1 we study the *optimal* performance that can be attained by CDPlm and the components it is composed of. To that end, we set λ_{clust} and λ_{psg} to values that yield optimized performance on a *per-query* basis. This practice enables to compare the relative effectiveness of whole-document-, passage-, and cluster-based information when completely neutralizing free-parameter-values effects. Then, in Sect. 4.3.2 we set λ_{clust} and λ_{psg} to values that result in optimized *average performance* over the set of queries for corpus. Doing so helps to shed light on the potential performance of CDPlm when using the same (effective) parameter values for all queries. Finally, in Sect. 4.3.3 we present performance numbers when *learning* the values of the free parameters of CDPlm, and those of the reference comparisons, using a leave-one-out cross-validation procedure performed over queries.

The evaluation metric for which performance is optimized in all cases is p@5.[7] The values of λ_{clust} and λ_{psg} are chosen from $\{0, 0.1, \ldots, 1\}$. For compatibility, we also use in Sect. 4.3.4 a leave-one-out cross validation procedure to train/test the learning-to-re-rank method, CDPsvm.

Efficiency considerations We segment documents to passages *prior* to retrieval time. Hence, the main computational overhead posted by our methods on top of the initial retrieval is *clustering* the initially retrieved list; specifically, computing inter-document similarities. However, the initial list is quite short—composed of only 50 documents—and therefore, this overhead is not substantial. Furthermore, inter-document-similarities could be computed based on *snippets* of documents, rather than using whole-document content, as was done for example, in work on clustering the results of Web search engines (Zamir and Etzioni 1998). Similar efficiency considerations were echoed in previous work on using query-specific clusters—i.e., clusters of top-retrieved documents—for re-ranking (Willett 1985; Liu and Croft 2004, 2006a; Kurland and Lee 2005, 2006; Yang et al. 2006) and, in work on graph-based re-ranking methods that utilize inter-document-similarities among top-retrieved documents (Diaz 2005; Kurland and Lee 2005; Krikon et al. 2009).

4.3 Experimental results

In what follows we present and analyze the performance of CDPlm and its components (Sects. 4.3.1–4.3.3), and that of CDPsvm (Sect. 4.3.4), when re-ranking an initial list that was retrieved using a language-model-based approach as described above. In Sect. 4.3.5 we study the effectiveness of CDPlm in re-ranking an initial list that was retrieved using Okapi-BM25 (Robertson et al. 1994).

4.3.1 Optimal-performance analysis

Our first order of business is studying the relative effectiveness of using whole-document-based, cluster-based, and passage-based information. To that end, we use free-parameter settings that yield specific instances of CDPlm [refer back to (6)]. Furthermore, we neutralize the effect of free parameters that are not fixed, by using values that yield optimal

[7] When optimizing performance per query, if two parameter configurations yield the same p@5, we choose the one maximizing p@10, as we are interested in optimal performance. When optimizing average performance over a set of queries per corpus, p@5 ties are broken by using the configuration *minimizing* p@10 so as to provide conservative estimates of performance. Finally, in case of p@5 ties in the learning phase of the cross validation procedure, we choose the configuration *maximizing* p@10 so as to *learn* the best possible configuration.

Table 2 Optimal-performance analysis of the information types utilized by CDPlm; free-parameter values are set to optimize per-query performance

	AP		FR		SJMN		TREC8		WSJ		WT10G	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
Upper bound	88.5	79.6	50.7	33.1	77.9	63.8	94.4	85.0	90.4	79.4	74.5	58.9
Doc (init. rank)	45.7	43.2	24.8	18.5	34.7	29.6	50.0	45.6	53.6	48.4	33.9	28.0
Clust	53.3^{i}_{c}	49.8^{i}_{c}	10.4^{i}_{c}	8.9^{i}_{c}	38.9_{c}	35.1^{i}_{c}	55.2_{c}	48.6_{c}	56.4_{c}	51.4_{c}	33.7_{c}	29.3_{c}
Psg	46.1_{c}	41.7_{c}	24.8_{c}	19.4_{c}	31.5^{i}_{c}	28.6_{c}	44.8^{i}_{c}	43.0_{c}	48.8_{c}	44.6_{c}	32.9_{c}	29.3_{c}
DocClust	61.0^{i}_{c}	55.6^{i}_{c}	28.1_{c}	20.6_{c}	49.1^{i}_{c}	40.4^{i}_{c}	66.0^{i}_{c}	54.4^{i}_{c}	64.8^{i}_{c}	56.8^{i}_{c}	44.3^{i}_{c}	36.1^{i}_{c}
DocPsg	53.3^{i}_{c}	47.9^{i}_{c}	30.4^{i}	21.3^{i}	38.9^{i}_{c}	34.0^{i}_{c}	53.6^{i}_{c}	49.2^{i}_{c}	58.8^{i}_{c}	51.6^{i}_{c}	41.2^{i}_{c}	34.3^{i}_{c}
ClustPsg	63.0^{i}_{c}	55.8^{i}_{c}	28.9_{c}	20.9_{c}	46.8^{i}_{c}	39.6^{i}_{c}	65.2^{i}_{c}	53.8^{i}_{c}	64.4^{i}_{c}	55.8^{i}_{c}	45.9^{i}_{c}	37.0^{i}_{c}
CDPlm	$\mathbf{65.9}^{i}$	$\mathbf{57.8}^{i}$	$\mathbf{32.6}^{i}$	$\mathbf{22.8}^{i}$	$\mathbf{50.6}^{i}$	$\mathbf{42.1}^{i}$	$\mathbf{68.0}^{i}$	$\mathbf{56.4}^{i}$	$\mathbf{66.8}^{i}$	$\mathbf{58.6}^{i}$	$\mathbf{49.6}^{i}$	$\mathbf{38.8}^{i}$

The best performance attained by a re-ranking method per corpus and evaluation measure is boldfaced; 'i' and 'c' mark statistically significant differences with Doc (the initial ranking) and CDPlm, respectively

p@5 on a per-query basis, as explained above. Such practice enables a fair comparison of the *optimal performance* that can be attained by CDPlm and its components. The parameter settings are:

- **Doc** ($\lambda_{clust} = \lambda_{psg} = 0$): the initial ranking that is based solely on whole-document information;
- **Clust** ($\lambda_{clust} = 1$): uses only cluster-based information; this is a previously proposed cluster-based (re-)ranking method (Kurland and Lee 2004; Kurland 2009);
- **Psg** ($\lambda_{clust} = 0$, $\lambda_{psg} = 1$): a commonly used method that utilizes only passage-based information (Callan 1994; Liu and Croft 2002);
- **DocClust** ($\lambda_{psg} = 0$): uses document-based and cluster-based information, and was shown to yield state-of-the-art re-ranking performance (Kurland 2009);[8]
- **DocPsg** ($\lambda_{clust} = 0$): uses document-based and passage-based information; this is also a commonly used passage-based ranking approach (Buckley et al. 1994; Callan 1994; Wilkinson 1994; Cai et al. 2004; Bendersky and Kurland 2008); and,
- **ClustPsg** ($\lambda_{psg} = 1$): uses cluster-based and passage-based information.

Table 2 presents the performance numbers. The numbers in the first row represent the upper bound on performance; that is, the performance attained by positioning all relevant documents in the initial list, $\mathcal{D}_{\text{init}}$, at the highest ranks.

Our first observation based on Table 2 is that when used alone, cluster-based information (Clust) is in most cases more effective than whole-document-based (Doc) and passage-based (Psg) information. The notable exception is the FR corpus for which Clust posts poor performance in comparison to that of Doc and Psg. This finding can be explained by the statistics presented in Table 1 about FR containing long and heterogeneous documents as manifested in within-document inter-passage similarities. As clustering is based on inter-document

[8] This method was originally termed "interpolation" (Kurland and Lee 2004; Kurland 2009).

similarities, and those could be dominated by many non-query-related aspects in case the documents are highly heterogeneous, clusters then convey less effective information for re-ranking than in cases wherein documents are relatively "homogeneous".[9] Indeed, WT10G, which also contains heterogeneous documents (refer to Table 1), is the second corpus in addition to FR, for which Clust underperforms Doc (in terms of p@5); for the news-based corpora, which contain relatively short and homogeneous documents, this does not happen. Nevertheless, we can see that integrating cluster-based information with whole-document-based (DocClust) or passage-based information (ClustPsg) yields effective re-ranking performance even for the FR and WT10G corpora.

More generally, the integration of any two types of information yields performance that is substantially better than that of using each alone; furthermore, the resultant performance is much better than that of the initial ranking. Specifically, the ClustPsg method outperforms both Clust and Psg by a considerable margin. As the performance of Psg is often below that of the initial document-based ranking, and that of Clust is often beyond that of the initial ranking, we conclude that passage-based and cluster-based information are complementary, and there are clear merits in integrating them.

We can also see in Table 2 that the performance of CDPlm, which integrates whole-document-, cluster-, and passage-based information, is better to a substantial (and often to a statistically significant) degree than that of its specific instances that utilize one or two of the three information types. Thus, the overall picture rising from Table 2 is that the integration of whole-document-, cluster-, and, passage-based information has a clear potential. In other words, if we were able to automatically set per each query the λ_{clust} and λ_{psg} parameters, which control the relative impact of the information types, to highly effective values, then the integration of these information types would be of clear merit. Still, there is much room for improvement, as the "upper bound" numbers attest, and which can be addressed by using some of the approaches discussed in Sect. 2 in addition to CDPlm.

4.3.2 *Performance analysis when using the same effective free-parameter values for all queries*

The analysis presented above focused on the optimal potential performance of CDPlm and its components. The optimal performance was attained by setting free-parameter values to optimize performance per each query. We now turn to analyze the potential effectiveness of CDPlm when using the same (effective) free-parameter values for all queries per corpus; specifically, λ_{clust} and λ_{psg} are set to values that optimize *average* (over queries) p@5. Naturally, finding such effective parameter values is a task at its own right, which we address in the next section using cross validation. Yet, such practice enables us to study, using a setup more practical than that above, the relative benefits of cluster-based and passage-based information. Furthermore, we can contrast the performance of CDPlm with that of reference comparisons when (partially) ameliorating the effects of free-parameter values, yet avoiding per-query fitting of parameter values.

We first study the general effectiveness of CDPlm as a re-ranking method. To that end, we compare its performance with that of the initial ranking upon which re-ranking is performed. Recall that the initial ranking was created by a standard language-model approach ($p_d(q)$) wherein the smoothing parameter, μ, was optimized for MAP. Hence, we

[9] While previous work showed that Psg substantially outperforms Doc when ranking the *entire* FR corpus (Callan 1994; Liu and Croft 2002; Bendersky and Kurland 2008), Table 2 shows that this is not the case when Psg is used to re-rank the list retrieved by Doc.

Table 3 Comparison with the initial ranking and optimized baselines when using the same (optimized) free-parameter values for all queries

	AP		FR		SJMN		TREC8		WSJ		WT10G	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
init. rank	45.7	43.2	24.8	18.5	34.7	29.6	50.0	45.6	53.6	48.4	33.9	28.0
opt. base	46.5^{i}	43.9	24.8	18.9	34.9	30.5^{i}	51.2	46.4	56.0	49.4	34.1	28.2
CDPlm	$\mathbf{54.3}^{i}_{o}$	$\mathbf{50.2}^{i}_{o}$	**26.7**	**20.0**	$\mathbf{42.3}^{i}_{o}$	$\mathbf{36.6}^{i}_{o}$	$\mathbf{57.6}^{i}$	**49.0**	**57.2**	**51.2**	**37.4**	$\mathbf{31.8}^{i}_{o}$

'i' and 'o' mark statistically significant differences with the former and latter, respectively. The best result in a column is boldfaced

also compare CDPlm with **optimized baselines** that use $p_d(q)$ to rank *all* documents in the corpus, with μ optimized for p@5 and p@10, independently. We can see in Table 3 that CDPlm consistently and substantially outperforms both the initial ranking and the optimized baselines, often, to a statistically significant degree.[10]

To further study the impact of cluster-based and passage-based information, we present in Fig. 1 the effect of varying the values of λ_{clust} and λ_{psg} on the p@5 performance of CDPlm; when setting one of the two parameters to some value, the value of the second parameter is set so as to maximize *average* (over queries) p@5. It is important to note that while λ_{clust} solely determines the impact of cluster-based information, both λ_{psg} and λ_{clust} determine that of passage-based information. [Refer back to (6)].

Putting aside the case for the FR corpus, we can see in Fig. 1 that the performance of CDPlm is much superior to that of the initial ranking for a vast majority of the values of λ_{clust} ($\neq 0$), and for all values of λ_{psg}. These findings attest to the merits of the way CDPlm utilizes and integrates passage-based and cluster-based information. Furthermore, we can see that using $\lambda_{clust} \in \{0.1, 0.2\}$ and $\lambda_{psg} \in \{0.2, 0.3\}$ often yields near-optimal performance.

For the FR corpus, we see as shown above, that using cluster-based information results in in-effective re-ranking performance. Furthermore, only relatively large values of λ_{psg}—i.e., putting a lot of emphasis on passage-based information—yield performance that is (much) better than that of the initial ranking. (For $\lambda_{psg} = 1$, no document-based information is used on top of passage-based information, and hence, there is a relative decrease in performance.) Similarly, we see that putting too much emphasis on cluster-based information is not effective for WT10g, which as FR, contains long and heterogeneous documents.

Comparison with pseudo-feedback-based query expansion The CDPlm method uses information from the initial list, $\mathcal{D}_{\text{init}}$, to re-rank it. Pseudo-feedback-based query expansion methods, on the other hand, use information from $\mathcal{D}_{\text{init}}$ to construct a query model using which the *entire* corpus is re-ranked. Furthermore, CDPlm, as noted above, can conceptually be viewed as integrating different approaches for representing a document, while query expansion methods focus on the query representation. Thus, we turn to compare the performance of CDPlm with that of a state-of-the-art pseudo-feedback-based query expansion approach, namely, *relevance model number 3* (**RM3**) (Lavrenko and Croft

[10] While our focus here is on precision at top ranks, we note that the MAP (optimized) performance of CDPlm at cutoff 50—the size of the initial list $\mathcal{D}_{\text{init}}$—also consistently transcends that of the initial ranking: CDPlm yields MAP of 10.1, 24.9, 15.9, 17.9, 23.1 and 14.0 over AP, FR, SJMN, TREC8, WSJ and WT10G, respectively; the initial ranking MAP is 9.3, 24.8, 14.6, 17.5, 22.3 and 13.3, respectively.

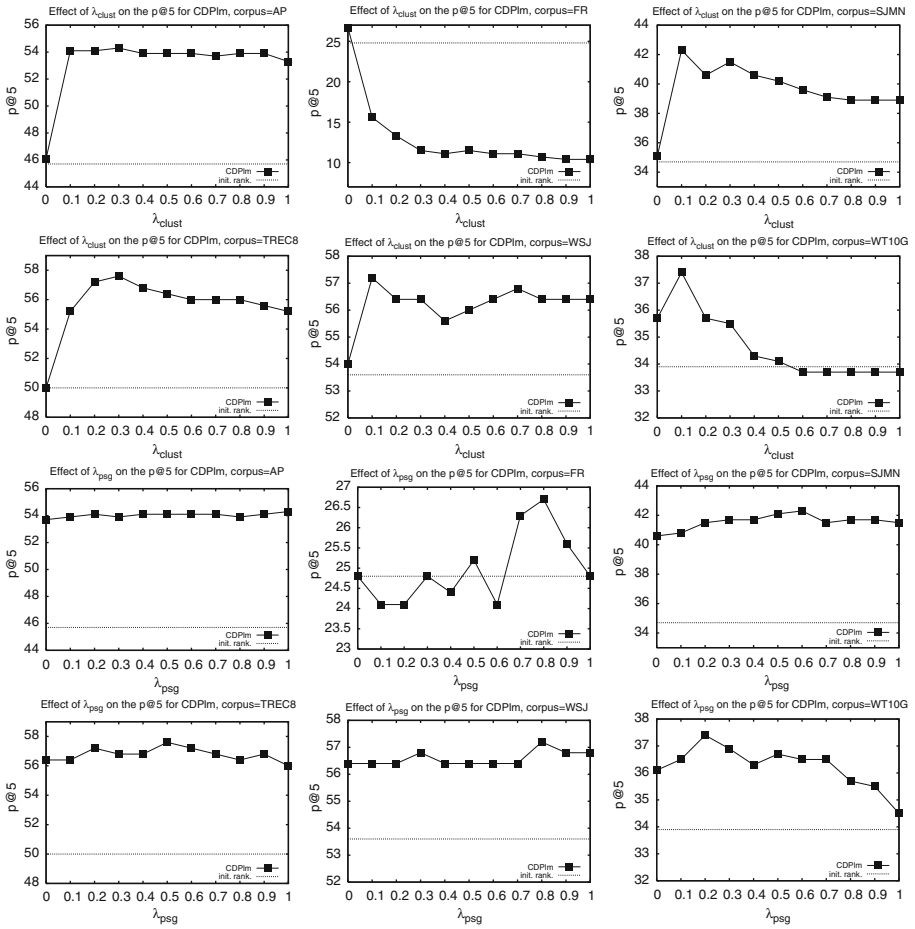

Fig. 1 Effect of varying λ_{clust} (*first* and *second* rows) and λ_{psg} (*third* and *fourth* rows) on the p@5 performance of CDPlm. The performance of the initial ranking, depicted with horizontal lines, is presented for reference. *Note*: figures are not to the same scale

2001; Abdul-Jaleel et al. 2004). For completeness of comparison, we also study a variant, **RM3(re)**, which uses the constructed relevance model to *re-rank* $\mathcal{D}_{\text{init}}$, rather than to rank the entire corpus. Ranking with a relevance model is based on its cross entropy with the document language model (Lavrenko 2004).

The values of the free parameters of RM3 and RM3(re) are set to optimize average p@5 over the set of queries per corpus, as is the case for CDPlm. Specifically, the (Jelinek–Mercer) smoothing parameter used for relevance-model construction is chosen from $\{0, 0.1, 0.3, \ldots .0.9\}$; the number of terms used by the models is chosen from {25, 50, 75, 100, 500, 1000, 5000, *ALL*}, where "*ALL*" stands for using all terms in the vocabulary; and, the interpolation parameter that controls the reliance on the original query is set to a value in $\{0, 0.1, \ldots, 0.9\}$. The (Dirichlet) document language model smoothing parameter (μ) used for ranking with a relevance model is set to 2000 as in all other methods. Table 4 presents the performance comparison.

Table 4 Comparison with a relevance model used to either rank all corpus (RM3) or to re-rank the initial list (RM3(re))

	AP		FR		SJMN		TREC8		WSJ		WT10G	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
init. rank	45.7	43.2	24.8	18.5	34.7	29.6	50.0	45.6	53.6	48.4	33.9	28.0
RM3	50.3^{i}	48.6^{i}	25.2	18.1	$\mathbf{42.3}^{i}$	$\mathbf{37.8}^{i}$	54.4	**49.4**	58.4^{i}	51.0	35.7	28.7
RM3(re)	51.1^{i}	48.2^{i}	25.2	18.1	41.5^{i}	35.7^{i}	54.4	48.6	$\mathbf{58.8}^{i}$	**52.0**	36.3	29.4
CDPlm	$\mathbf{54.3}^{i}_{r}$	$\mathbf{50.2}^{i}$	**26.7**	**20.0**	$\mathbf{42.3}^{i}$	36.6^{i}	$\mathbf{57.6}^{i}$	49.0	57.2	51.2	**37.4**	$\mathbf{31.8}^{i}_{r}$

Free-parameter values are set for each method so as to optimize average p@5 per corpus. Best result in a column is boldfaced; 'i' and 'r' mark statistically-significant differences with the initial ranking and RM3, respectively; the differences between CDPlm and RM3(re) are not statistically significant

We can see that the performance of CDPlm is superior to that of the relevance models in most relevant comparisons (corpus × evaluation measure). Specifically, CDPlm posts p@5 performance—the metric for which performance was optimized—that is substantially better than that of the relevance models over AP and TREC8; the improvement over RM3 for AP is also statistically significant. We can also see that in the few cases that CDPlm is outperformed by the relevance models the performance differences are not statistically significant.

4.3.3 Learning free-parameter values

Heretofore, we evaluated the potential performance of CDPlm, and that of the reference comparisons, by ameliorating issues that rise from free-parameter values. Now, we turn to study whether effective parameter values generalize from one query to another. We note that this study is different than that presented in Fig. 1, wherein we analyzed the robustness of the average (over queries) performance of CDPlm with respect to free-parameter values.

We take a leave-one-out cross-validation approach. The free-parameter values of a method per query are set to those optimizing p@5 performance over all other queries for

Table 5 Performance numbers when learning free parameter values using a leave-one-out cross validation procedure

	AP		FR		SJMN		TREC8		WSJ		WT10G	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
init. rank	45.7	43.2	24.8	18.5	34.7	29.6	50.0	45.6	53.6	48.4	33.9	28.0
DocPsg	38.8^{i}	41.1	**25.9**	**20.2**	34.0	30.2	48.0^{i}	44.8	54.0	48.8	35.5	29.5
DocClust	$\mathbf{53.3}^{i}_{p}$	$\mathbf{50.2}^{i}_{p}$	24.8	18.5	38.9_{p}	36.0^{i}_{p}	53.2	47.4	53.6	50.8	36.1	$\mathbf{31.8}^{i}$
RM3	49.9_{p}	48.5^{i}_{p}	23.7	18.1	41.5^{i}_{p}	$\mathbf{37.4}^{i}_{p}$	50.8	**49.4**	**54.8**	**51.0**	30.4^{i}_{pc}	28.1_{c}
RM3(re)	51.1^{i}_{p}	48.2^{i}_{p}	23.7	18.1	37.4_{r}	34.6^{i}_{pr}	50.4	47.6	52.0_{r}	50.6	36.3_{r}	29.4_{r}
CDPlm	52.3^{i}_{p}	49.2^{i}_{p}	**25.9**	**20.2**	$\mathbf{42.1}^{ie}_{pc}$	36.6^{ie}_{p}	**53.6**	49.0	52.8	50.6	$\mathbf{37.3}_{r}$	$\mathbf{31.8}^{i}_{r}$

'i', 'p', 'c', 'r', and 'e' mark statistically significant differences with init. rank, DocPsg, DocClust, RM3, and RM3(re), respectively. Boldface marks the best result in a column

the same corpus. We present the resultant performance of CDPlm and the reference comparisons in Table 5.

Our first observation based on Table 5 is that CDPlm outperforms the initial ranking in almost all reference comparisons; often, the improvements are substantial and statistically significant. This finding further attests to the effectiveness of CDPlm in re-ranking.

Another observation we make based on Table 5 is that CDPlm outperforms its specific instantiations, DocPsg (a standard passage-based ranking method (Buckley et al. 1994; Callan 1994; Wilkinson 1994; Cai et al. 2004; Bendersky and Kurland 2008)) and DocClust [a state-of-the-art cluster-based re-ranking approach (Kurland and Lee 2004; Kurland 2009)] in most relevant comparisons; in several cases (e.g., refer to AP and SJMN), the performance differences are also statistically significant. Furthermore, DocPsg and DocClust never outperform CDPlm in a statistically significant manner. These findings show that the relative importance of whole-document-, passage-, and cluster-based information, as determined by CDPlm's free-parameters' values, can be relatively effectively learned across queries. Naturally, however, the performance numbers (both for CDPlm and for DocPsg and DocClust) are much lower than those presented in Table 2, which were attained by setting parameter values so as to optimize per-query performance. Hence, there is much room for improvement that can potentially be obtained by devising methods for *automatically* setting the relative importance of whole-document, passage, and cluster-based information on a per-query basis.

We can also see in Table 5 that CDPlm outperforms RM3, which ranks the entire corpus, and RM3(re), which re-ranks the initial list, in most relevant comparisons; some of these performance differences are also statistically significant. We also note that while RM3 outperforms CDPlm over WSJ—although, not to a statistically significant degree—the statistically significant improvements posted by CDPlm over RM3 for WT10G are quite striking. As is the case for CDPlm, the relevance model approach can benefit much from devising methods for automatically setting free-parameter values on a per-query basis. A case in point, compare the performance numbers of the relevance-model implementations presented in Tables 4 and 5—the former, which in many cases are much better than the latter, are based on using free-parameter values that result in optimized average performance for a corpus, and the latter are based on using cross validation to set free-parameter values.

All in all, we see that in general, when learning free parameter values using cross validation, CDPlm is the most effective method among those presented in Table 5. (Note that the p@5—the metric based on which learning of free parameter values was performed—posted by CDPlm is the best for four out of six corpora; furthermore, CDPlm is the only method in Table 5 that is never outperformed in a statistically significant manner by any other method.)

4.3.4 Learning to re-rank

The learning-to-re-rank method, CDPsvm, uses SVM^{rank} (Joachims 2006). We use the default values for all SVM^{rank} parameters, except for that of c, which controls the bias-variance trade-off. As it turns out, c has considerable impact on the resultant re-ranking performance. Thus, we present performance numbers for two settings of CDPsvm.

The first setting, **CDPsvm(B)**, is based on using a leave-one-out cross validation for training/testing SVM^{rank} over *all* queries for each value of c. Then, the value of c that yields the best (average over queries) p@5 performance is selected, and the resultant performance is reported. In the second setting, **CDPsvm(L)**, the value of c is learned for *each query* as follows. We perform a leave-one-out cross-validation over the *rest* of the queries to find the value of c that optimizes p@5. Using this value, we then learn a model

Table 6 Comparison of CDPsvm with CDPlm

	AP		FR		SJMN		TREC8		WSJ		WT10G	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
init. rank	45.7	43.2	24.8	18.5	34.7	29.6	50.0	45.6	53.6	48.4	33.9	28.0
CDPlm(B)	**54.3**^{i}	**50.2**^{i}	**26.7**	**20.0**	**42.3**^{i}	**36.6**^{i}	**57.6**^{i}	49.0	**57.2**	51.2	**37.4**	**31.8**^{i}
CDPsvm(B)	**54.3**^{i}	49.8^{i}	24.4	19.1	41.5^{i}	36.4^{i}	57.2^{i}	**49.2**	56.4	**51.4**	36.5	29.7
CDPlm(L)	52.3^{i}	49.2^{i}	**25.9**	**20.2**	**42.1**^{i}	**36.6**^{i}	53.6	**49.0**	**52.8**	**50.6**	**37.3**	**31.8**^{i}
CDPsvm(L)	**54.3**$^{i}_{l}$	**49.9**^{i}	24.4	19.1	41.3^{i}	36.4^{i}	**55.6**	48.8	52.4	49.8	33.7$_{l}$	29.7$_{l}$

The best result in a block is boldfaced; 'l' marks statistically difference between CDPsvm(L) and CDPlm(L); the performance differences between CDPsvm(B) and CDPlm(B) are not statistically significant; 'i' marks statistically significant difference with the initial ranking

using these queries and apply it to the query at hand. The values of c are chosen from $\{10^{-5}, 5 * 10^{-5}, \ldots, 0.1, 0.5, 5, 50, 500, 1000, 5000, 10000\}$.

For comparison purposes, we present the performance of CDPlm when its two free parameters, λ_{clust} and λ_{psg} are optimized for average-over-queries performance (**CDPlm(B)**), as was the case in Table 3; and, its performance when using leave-one-out cross validation to learn the values of these parameters (**CDPlm(L)**), as was the case in Table 5.

We can see in Table 6 that in most reference comparisons, the implementations of our methods improve over the initial ranking, often to a substantial and statistically significant degree. This finding further supports the merits of integrating cluster-, document-, and passage-based information for re-ranking, whether using a probabilistic model (CDPlm) or a learning-to-rank approach (CDPsvm).

Evidently, the *potential* performance of CDPlm is somewhat better than that of CDPsvm as manifested in the best-parameter-values setups ('B') for most relevant comparisons. Now, recall that both CDPlm and CDPsvm use a linear interpolation of the same language-model-based estimates. Hence, these performance differences—although not statistically significant—may imply that learning a "good" balance between the three information types (cluster-, document-, and passage-based) in a discriminative manner by SVM^{rank} can fall short, possibly due to query-variability issues (Peng et al. 2010).

The comparison between CDPlm and CDPsvm when learning free parameter values ('L') reveals that the performance of the former is in most relevant comparisons somewhat superior to that of the latter; for WT10G, the difference is quite substantial and also statistically significant.

We can also see in Table 6 that in some cases the performance of CDPlm and CDPsvm can quite decrease when moving from the best ('B') to the learned ('L') parameter settings. Thus, while both CDPlm and CDPsvm are very effective in most reference comparisons when learning free-parameter values, there is still room for improvement with respect to setting these values on a per-query basis—a challenging task, as mentioned above, that we leave for future work.

4.3.5 Re-ranking an Okapi-BM25-based initially retrieved list

Insofar, the initial list, $\mathcal{D}_{\text{init}}$, upon which re-ranking was performed, was set to the 50 documents that were the highest ranked by a language-model-based approach. We now

Table 7 Performance of CDPlm when re-ranking an initial list of documents that was retrieved using Okapi-BM25

	AP		FR		SJMN		TREC8		WSJ		WT10G	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
BM25 init. rank	45.5	42.9	26.7	**21.1**	34.3	31.1	48.0	45.2	54.0	50.8	35.9	29.8
CDPlm	**52.5**^{b}	**48.9**^{b}	**27.0**	20.7	**42.1**^{b}	**35.4**^{b}	**55.6**^{b}	**47.2**	**58.8**	**53.2**	**37.8**	**33.0**^{b}

The best result in a column is boldfaced; 'b' marks statistically-significant difference with the BM25-based initial ranking

turn to study whether CDPlm is effective in re-ranking an initial list that is retrieved by a different retrieval method; specifically, we use Okapi-BM25 (Robertson et al. 1994). As was the case for the initial language-model-based ranking, we set Okapi's free-parameters to values that optimize MAP (@1000) so as to create an initial list of a reasonable quality. Following previous recommendations (Robertson et al. 2000, 2004), we use the following free-parameter values ranges: $k_1 \in \{0.1, 0.25, 0.5, 0.75, 0.9, 1, 1.2, 2, 2.5, 3\}$; $k_3 \in \{0.1, 0.2, 0.5, 0.8, 1, 2, 5, 7, 10, 15, 20\}$; and, $b \in \{0.1, 0.2, 0.3, 0.5, 0.75, 0.85, 0.95, 1, 1.5, 2.5, 3\}$. The 50 highest ranked documents are re-ranked using CDPlm, which uses language-model-based estimates as described above; CDPlm's free-parameter values are set to optimize average p@5 performance per corpus as was the case in Sect. 4.3.2 The performance numbers are presented in Table 7.

As we can see in Table 7, the performance of CDPlm is superior to that of the Okapi-based initial ranking in almost all reference comparisons; furthermore, in quite a few cases the improvements are statistically significant. These findings further demonstrate the effectiveness of CDPlm in re-ranking.

Note For the WSJ corpus the Okapi-BM25 initial ranking can be quite improved if stopwords are removed from queries. (We used Zettair's stopword list; recall that in our experimental setup above stopwords were not removed from queries and documents.) For the other corpora, however, it is the case that the improvements are smaller or there are no improvements or there can even be performance degradation. For WSJ removing stopwords from queries results in initial Okapi-BM25 ranking with p@5 and p@10 of 57.2 and 49.4, respectively. Employing CDPlm upon this initial ranking yields p@5 and p@10 of 61.2 and 54.2, respectively; the p@10 improvement is also statistically significant. Thus, we see that even when improving the effectiveness of the initial ranking (by using a different pre-processing regime here), CDPlm still posts quite substantial performance improvements over this ranking.

5 Conclusions and future work

Cluster-based and passage-based document ranking approaches could be viewed as employing two opposite approaches for document representation. Cluster-based document retrieval is often based on expanding the document representation with corpus context manifested in the clustering structure. Passage-based document retrieval is based on focusing on a specific part of the document.

We presented a study of the relative merits of each of these approaches, and of the potential of integrating them. To perform the study, we devised two retrieval methods that

integrate whole-document-, cluster-, and passage-based information. The first is a probabilistic approach that uses document-based, passage-based and cluster-based language models. The second is a discriminative, learning-to-rank, approach that uses language-model-based estimates.

We evaluated and studied the proposed methods when applied for the re-ranking task—re-ordering documents in an initially retrieved list so as to improve precision at the very top ranks. We showed that the methods consistently and substantially outperform the initial ranking. The resultant performance of the probabilistic approach also transcends that of document ranking methods that use either cluster-based or passage-based information, but not both. Hence, the empirical findings support the complementary nature of these two information types, and the potential in integrating them. Furthermore, we showed that the integration can yield performance that often transcends that of a state-of-the-art pseudo-feedback-based query expansion method—i.e., an approach that focuses on query representation, rather than on document representation, which is the focus of this paper.

In addition, the study showed that using cluster-based information is much more effective than using passage-based information for document ranking, except for corpora containing very large (and heterogeneous) documents for which the reverse holds. Nevertheless, integrating cluster-based and passage-based information can yield performance that substantially transcends that of using each alone. More generally, we showed that integrating these two types of information with whole-document-based information can yield performance that is substantially better than that of using any subset of the three information types.

A future direction that emerged in the study was devising an automatic way of balancing the use of whole-document-, cluster-, and passage-based information on a per-query basis. While there is some work on controlling the use of whole-document-based versus passage-based information (Bendersky and Kurland 2008) on a per-document basis, an open challenge is how to balance those with respect to using cluster-based information on a per-query basis.

As noted above, the study presented in this paper addresses one component of a search system; that is, (a part of) the document representation task is addressed from an *effectiveness* perspective. As already stated, our approach does not incur a considerable computational overhead over the initial ranking that is based on document-query similarities. Hence, from an *efficiency* point of view, the approach is applicable in practical retrieval settings. Yet, a natural question, which rises with regard to Cranfield-style-based evaluations (Hersh et al. 2000; Turpin and Hersh 2001; Smucker and Jethani 2010) as the one we presented here, is whether the presented effectiveness improvements can be translated to improved user satisfaction/effectiveness. While this is an interesting question at its own right, we note that there are still additional means that can be employed so as to potentially improve the performance of our approach, and which can further increase the potential for merits to the user in practical search settings. For example, while our focus was on the "document side", integrating in addition different (expanded) query representations can potentially help improve performance; e.g., cluster-based (and topic-model-based) query expansion (Liu and Croft 2004; Tao et al. 2006; Wei and Croft 2006; Kalmanovich and Kurland 2009) and passage-based query expansion (Liu and Croft 2002; Bendersky and Kurland 2008) were shown to be of merit. Furthermore, using different types of passages, and utilizing different types of language models and/or term-proximity-based models, can also potentially improve performance as mentioned in Sect. 2.

Acknowledgments We thank the reviewers for their helpful comments. This paper is based upon work supported in part by Israel's Science Foundation under grant no. 890015, and by IBM's SUR award. Any

opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsoring institutions.

References

Abdul-Jaleel, N., Allan, J., Croft, W. B., Diaz, F., Larkey, L., Li, X., et al. (2004). UMASS at TREC 2004—novelty and hard. In *Proceedings of TREC-13* (pp. 715–725).

Baliński, J., & Daniłowicz, C. (2005). Re-ranking method based on inter-document distances. *Information Processing and Management, 41*(4), 759–775.

Beigbeder, M., Imafouo, A., & Mercier, A. (2009). Ensm-se at inex 2009: Scoring with proximity and semantic tag information. In *Proceedings of INEX* (pp. 49–58).

Bendersky, M., & Kurland, O. (2008). Utilizing passage-based language models for document retrieval. In *Proceedings of ECIR* (pp. 162–174).

Buckley, C., Salton, G., Allan, J., & Singhal, A. (1994). Automatic query expansion using SMART: TREC3. In *Proceedings of TREC-3* (pp. 69–80).

Cai, D., Yu, S., Wen, J.-R., & Ma, W.-Y. (2004). Block-based web search. In *Proceedings of SIGIR* (pp. 456–463).

Callan, J. P. (1994). Passage-level evidence in document retrieval. In *Proceedings of SIGIR* (pp. 302–310).

Craswell, N., Hawking, D., & Thistlewaite, P. B. (1999). Merging results from isolated search engines. In *Proceedings of the Australian Database Conference* (pp. 189–200).

Croft, W. B. (1980). A model of cluster searching based on classification. *Information Systems, 5*, 189–195.

Croft, W. B. (Ed.). (2000a). *Advances in information retrieval: Recent research from the center for intelligent information retrieval*. No. 7 in The Kluwer International Series on Information Retrieval. Kluwer.

Croft, W. B. (2000b). Combining approaches to information retrieval. In *Croft (2000a)*, No. 7 in The Kluwer International Series on Information Retrieval, Ch. 1 (pp. 1–36).

Croft, W. B., & Lafferty, J. (Eds.) (2003). *Language modeling for information retrieval*. No. 13 in Information Retrieval Book Series. Kluwer.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.

Denoyer, L., Zaragoza, H., & Gallinari, P. (2001). HMM-based passage models for document classification and ranking. In *Proceedings of ECIR* (pp. 126–135).

Diaz, F. (2005). Regularizing ad hoc retrieval scores. In *Proceedings of CIKM* (pp. 672–679).

Diaz, F. (2008). A method for transferring retrieval scores between collections with non overlapping vocabularies. In *Proceedings of SIGIR* (pp. 805–806), poster.

Elsas, J. L., & Dumais, S. T. (2010). Leveraging temporal dynamics of document content in relevance ranking. In *Proceedings of WSDM* (pp. 1–10).

Fisher, H. L., & Elchesen, D. R. (1972). Effectiveness of combining title words and index terms in machine retrieval searches. *The Computer Journal, 35*(3), 243–255.

Griffiths, A., Luckhurst, H. C., & Willett, P. (1986). Using interdocument similarity information in document retrieval systems. *Journal of the American Society for Information Science (JASIS), 37*(1), 3–11.

Hearst, M. A., & Plaunt, C. (1993). Subtopic structuring for full-length document access. In *Proceedings of SIGIR* (pp. 56–89).

Hersh, W. R., Turpin, A., Price, S., Chan, B., Kraemer, D., Sacherek, L., et al. (2000). Do batch and user evaluation give the same results? In *Proceedings of SIGIR* (pp. 17–24).

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

Hussain, M. (2004). *Language modeling based passage retrieval for question answering systems*. Master's thesis, Saarland University.

Ingwersen, P. (1994). Polyrepresentation of information needs and semantic entities: Elements of a cognitive theory for information retrieval interaction. In *Proceedings of SIGIR* (pp. 101–110).

Jardine, N., & van Rijsbergen, C. J. (1971). The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval, 7*(5), 217–240.

Jiang, J., & Zhai, C. (2004). UIUC in HARD 2004—passage retrieval using HMMs. In *Proceedings of the 13th text retrieval conference (TREC-13)*.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of SIGKDD* (pp. 133–142).

Joachims, T. (2006). Training linear SVMs in linear time. In *Proceedings of KDD* (pp. 217–226).

Joyce, T., & Needham, R. M. (1958). The thesaurus approach to information retrieval. *American Documentation, 9*(3), 192–197.

Kalmanovich, I. G., & Kurland, O. (2009). Cluster-based query expansion. In *Proceedings of SIGIR* (pp. 646–647), poster.

Kaszkiel, M., & Zobel, J. (1997). Passage retrieval revisited. In *Proceedings of SIGIR* (pp. 178–185).

Kaszkiel, M., & Zobel, J. (2001). Effective ranking with arbitrary passages. *Journal of the American Society for Information Science, 52*(4), 344–364.

Katzer, J., McGill, M., Tessier, J., Frakes, W., & Dasgupta, P. (1982). A study of the overlap among document representations. *Information Technology: Research and Development, 1*(2), 261–274.

Kraaij, W., Westerveld, T., & Hiemstra, D. (2002). The importance of prior probabilities for entry page search. In *Proceedings of SIGIR* (pp. 27–34).

Krikon, E., Kurland, O., & Bendersky, M. (2009). Utilizing inter-passage and inter-document similarities for re-ranking search results. In *Proceedings of CIKM* (pp. 1597–1600).

Kurland, O. (2006). *Inter-document similarities, language models, and ad hoc retrieval*. Ph.D. thesis, Cornell University.

Kurland, O. (2009). Re-ranking search results using language models of query-specific clusters. *Journal of Information Retrieval, 12*(4), 437–460.

Kurland, O., & Lee, L. (2004). Corpus structure, language models, and ad hoc information retrieval. In *Proceedings of SIGIR* (pp. 194–201).

Kurland, O., & Lee, L. (2005). PageRank without hyperlinks: Structural re-ranking using links induced by language models. In *Proceedings of SIGIR* (pp. 306–313).

Kurland, O., & Lee, L. (2006). Respect my authority! HITS without hyperlinks utilizing cluster-based language models. In *Proceedings of SIGIR* (pp. 83–90).

Kwok, K. L. (1975). The use of title and cited titles as document representation for automatic classification. *Informationg Processing and Management, 11*(12), 201–206.

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

Lavrenko, V. (2004). *A generative theory of relevance*. Ph.D. thesis, University of Massachusetts Amherst.

Lavrenko, V., Allan, J., DeGuzman, E., LaFlamme, D., Pollard, V., & Thomas, S. (2002). Relevance models for topic detection and tracking. In *Proceedings of the human language technology conference (HLT)* (pp. 104–110).

Lavrenko, V., & Croft, W. B. (2001). Relevance-based language models. In *Proceedings of SIGIR* (pp. 120–127).

Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval, 3*(3), 225–331.

Liu, X., & Croft, W. B. (2002). Passage retrieval based on language models. In *Proceedings of CIKM* (pp. 375–382).

Liu, X., & Croft, W. B. (2004). Cluster-based retrieval using language models. In *Proceedings of SIGIR* (pp. 186–193).

Liu, X., & Croft, W. B. (2006a). *Experiments on retrieval of optimal clusters*. Tech. Rep. IR-478, Center for Intelligent Information Retrieval (CIIR), University of Massachusetts.

Liu, X., & Croft, W. B. (2006b). Representing clusters for retrieval. In *Proceedings of SIGIR* (pp. 671–672), poster.

Liu, X., & Croft, W. B. (2008). Evaluating text representations for retrieval of the best group of documents. In *Proceedings of ECIR* (pp. 454–462).

Lv, Y., & Zhai, C. (2009). Adaptive relevance feedback in information retrieval. In *Proceedings of CIKM* (pp. 255–264).

McBryan, O. A. (1994). GENVL and WWWW: Tools for taming the Web. In *Proceedings of WWW*.

McGill, M., Koll, M., & Noreault, T. (1979). *An evaluation of factors affecting document ranking by information retrieval systems*. Final report for grant nsf-ist-78-10454 to the National Science Foundation. Tech. rep., Syracuse University.

Meister, L., Kurland, O., & Kalmanovich, I. G. (2010). Re-ranking search results using an additional retrieved list. *Information Retrieval*.

Metzler, D., & Croft, W. B. (2005). A Markov random field model for term dependencies. In *Proceedings of SIGIR* (pp. 472–479).

Metzler, D., Novak, J., Cui, H., & Reddy, S. (2009). Building enriched document representations using aggregated anchor text. In *Proceedings of SIGIR* (pp. 219–226).

Mittendorf, E., & Schäuble, P. (1994). Document and passage retrieval based on hidden Markov models. In *Proceedings of SIGIR* (pp. 318–327).

Murdock, V., & Croft, W. B. (2005). A translation model for sentence retrieval. In *Proceedings of HLT/EMNLP* (pp. 684–695).
Na, S.-H., Kang, I.-S., Lee, Y.-H., & Lee, J.-H. (2008). Completely-arbitrary passage retrieval in language modeling approach. In *Proceedings of AIRS* (pp. 22–33).
Ogilvie, P., & Callan, J. (2003). Combining document representations for known item search. In *Proceedings of SIGIR* (pp. 143–150).
Peng, J., Macdonald, C., & Ounis, I. (2010). Learning to select a ranking function. In *ECIR* (pp. 114–126).
Ponte, J. M., & Croft, W. B. (1997). Text segmentation by topic. In *Proceedings of the European Conference on research and advanced technology for digital libraries* (pp. 113–125).
Ponte, J. M., & Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of SIGIR* (pp. 275–281).
Radev, D. R., Hovy, E. H., & McKeown, K. (2002). Introduction to the special issue on summarization. *Computational Linguistics, 28*(4), 399–408.
Robertson, S., Zaragoza, H., & Taylor, M. (2004). Simple BM25 extension to multiple weighted fields. In *Proceedings CIKM* (pp. 42–49).
Robertson, S. E., Walker, S., & Hancock-Beaulieu, M. (2000). Experimentation as a way of life: Okapi at trec. *Information Processing and Management, 36*(1), 95–108.
Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M., & Gatford, M. (1994). Okapi at trec-3. In *Proceedings of TREC-3*.
Salton, G. (1963). Associative document retrieval techniques using bibliographic information. *Journal of the ACM, 10*(4), 440–457.
Salton, G., Allan, J., & Buckley, C. (1993). Approaches to passage retrieval in full text information systems. In *Proceedings of SIGIR* (pp. 49–58).
Salton, G., & Buckley, C. (1991). Automatic text structuring and retrieval-experiments in automatic encyclopedia searching. In *Proceedings of SIGIR* (pp. 21–30).
Salton, G., & Lesk, M. (1968). Computer evaluation of indexing and text processing. *Journal of the ACM, 15*(1), 8–36.
Salton, J., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613–620.
Sanderson, M., & Zobel, J. (2005). Information retrieval system evaluation: Effort, sensitivity and reliability. In *Proceedings of SIGIR* (pp. 162–169).
Singhal, A., & Pereira, F. (1999). Document expansion for speech retrieval. In *Proceedings of SIGIR* (pp. 34–41).
Smucker, M. D., Allan, J., & Carterette, B. (2007). A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of CIKM* (pp. 623–632).
Smucker, M. D., & Jethani, C. P. (2010). Human performance and retrieval precision revisited. In *Proceedings of SIGIR* (pp. 595–602).
Tao, T., Wang, X., Mei, Q., & Zhai, C. (2006). Language model information retrieval with document expansion. In *Proceedings of HLT/NAACL* (pp. 407–414).
Tombros, A., & Sanderson, M. (1998). Advantages of query biased summaries in information retrieval. In *Proceedings of SIGIR* (pp. 2–10).
Turpin, A., & Hersh, W. R. (2001). Why batch and user evaluations do not give the same results. In *Proceedings of SIGIR* (pp. 225–231).
Van, T.-T., & Beigbeder, M. (2008). A comparison of re-ranking methods in digital libraries using user profiles. In *Proceedings of web intelligence* (pp. 751–754).
Voorhees, E. M. (1985). The cluster hypothesis revisited. In *Proceedings of SIGIR* (pp. 188–196).
Voorhees, E. M. (2002). Overview of the TREC 2002 question answering track. In *The 11th text retrieval conference TREC-11* (pp. 115–123).
Voorhees, E. M. (2005). Overview of the TREC 2005 robust retrieval task. In *Proceedings of TREC-14*.
Voorhees, E. M., & Harman, D. K. (2005). *TREC: Experiments and evaluation in information retrieval*. Cambridge: The MIT Press.
Wade, C., & Allan, J. (2005). *Passage retrieval and evaluation*. Tech. Rep. IR-396, Center for Intelligent Information Retrieval (CIIR), University of Massachusetts.
Wang, M., & Si, L. (2008). Discriminative probabilistic models for passage based retrieval. In *Proceedings of SIGIR* (pp. 419–426).
Wei, X., & Croft, W. B. (2006). LDA-based document models for ad-hoc retrieval. In *Proceedings of SIGIR* (pp. 178–185).
Wilkinson, R. (1994). Effective retrieval of structured documents. In *Proceedings of SIGIR* (pp. 311–317).
Willett, P. (1985). Query specific automatic document classification. *International Forum on Information and Documentation, 10*(2), 28–32.

Xu, J., & Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of SIGIR* (pp. 4–11).

Yang, L., Ji, D., Zhou, G., Nie, Y., & Xiao, G. (2006). Document re-ranking using cluster validation and label propagation. In *Proceedings of CIKM* (pp. 690–697).

Yi, X., & Allan, J. (2009). A comparative study of utilizing topic models for information retrieval. In *Proceedings of ECIR* (pp. 29–41).

Yue, Y., Finley, T., Radlinski, F., & Joachims, T. (2007). A support vector method for optimizing average precision. In *Proceedings of SIGIR* (pp. 271–278).

Zamir, O., & Etzioni, O. (1998). Web document clustering: A feasibility demonstration. In *Proceedings of SIGIR* (pp. 46–54).

Zhai, C., & Lafferty, J. D. (2001a). Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of CIKM* (pp. 403–410).

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

Zhao, J., & Yun, Y. (2009). A proximity language model for information retrieval. In *Proceedings of SIGIR* (pp. 291–298).