Learning Asymmetric Co-Relevance

Fiana Raiber
Technion, Israel
fiana@tx.technion.ac.il

Oren Kurland
Technion, Israel
kurland@ie.technion.ac.il

Filip Radlinski
Microsoft Cambridge, UK
filiprad@microsoft.com

Milad Shokouhi
Microsoft Cambridge, UK
milads@microsoft.com

ABSTRACT

Several applications in information retrieval rely on asymmetric co-relevance estimation; that is, estimating the relevance of a document to a query under the assumption that another document is relevant. We present a supervised model for learning an asymmetric co-relevance estimate. The model uses different types of similarities with the assumed relevant document and the query, as well as document-quality measures. Empirical evaluation demonstrates the merits of using the co-relevance estimate in various applications, including cluster-based and graph-based document retrieval. Specifically, the resultant performance transcends that of using a wide variety of alternative estimates, mostly symmetric inter-document similarity measures that dominate past work.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models

Keywords: asymmetric co-relevance

1. INTRODUCTION

The cluster hypothesis states that *closely associated documents tend to be relevant to the same requests* [13, 38]. One operational interpretation of the hypothesis is that inter-document similarities can serve as *symmetric co-relevance* estimates. That is, given two highly similar documents, they will either both be relevant, or not, to any given query. Many cluster-based (e.g., [13, 40, 18, 26, 28, 16, 17, 33]) and some graph-based [49, 7, 20] retrieval methods implicitly assume the correctness of this hypothesis.

However, a number of retrieval methods and applications rely, by design, on estimating *asymmetric co-relevance* [40, 11, 7, 16, 29, 20, 33], namely the relevance of a document to a query given another document assumed to be relevant. For example, nearest-neighbor clustering of top-retrieved documents is used in many cluster-based document retrieval methods, where each cluster consists of a document and its nearest neighbors in the similarity space [28, 16, 33]. These similarities presumably reflect asymmetric co-relevance. Voorhees' nearest-neighbor cluster hypothesis test is another example [40]: a relevant document is fixed and its co-relevant documents are sought. Yet another example is graph-based re-ranking methods, where the weight of a *directed* edge connecting two documents should reflect the probability that the target document is relevant assuming that the source document is relevant [20]. Despite all these applications, there has been very little work on devising asymmetric co-relevance estimates [49, 20]. Often, symmetric inter-document similarities are simply used instead [11, 28, 29].

We propose a novel model to learn an asymmetric co-relevance estimate. Most retrieval methods that rely on co-relevance estimates operate on a result list of the documents most highly ranked by initial search [16, 29, 20, 33]. Hence, our model is applied to documents in *some* initial result list.

The proposed model estimates co-relevance to a given document in the result list by utilizing rankings of documents in the list. The rankings are produced using a variety of functions. Some of the functions are based on the similarity with the query, the given document, or both. Several functions utilize passage-based document representations. Others utilize query-independent document-quality measures. We then train a supervised model that produces a co-relevance estimate by fusing the induced rankings. Given the ranking functions used, the learned estimate is inherently asymmetric.

Our experimental results show that the proposed estimate substantially outperforms numerous previously proposed co-relevance estimates in a variety of applications, including several state-of-the-art cluster-based and graph-based document retrieval methods.

Our main contributions can be summarized as follows.

- Presenting a novel supervised model for learning an asymmetric co-relevance estimate.
- Showing that using the proposed estimate in several state-of-the-art retrieval methods yields significant performance improvements over a wide variety of alternative estimates.

2. RELATED WORK

Numerous symmetric query-independent inter-document similarity measures were proposed for estimating co-relevance, including [2, 7, 20, 45]. Since documents co-relevant to one query are not necessarily co-relevant to another query [12], symmetric query-biased inter-document similarities have also

ICTIR'15, September 27–30, Northampton, MA, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3833-2/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2808194.2809454 .

been proposed [37, 30]. Our asymmetric co-relevance estimation approach integrates both query-biased and query-independent inter-document similarity measures, and leads to better performance in several retrieval methods and applications compared to using these measures separately.

The assumption underlying the optimum clustering framework is that documents should be deemed similar if they are co-relevant to many queries [10]. In contrast, one of the assumptions on which our method relies is that documents should be deemed co-relevant if they are similar given many ranking functions. We show that our method outperforms an inter-document similarity measure instantiated from the optimum clustering framework in various applications.

Beyond the document level, several inter-document similarity measures utilize query-independent passage-based similarities [41, 42, 32]. Inter-passage similarities were also used for document retrieval [15]. We also include several such methods in our supervised model [32], thereby benefiting from passage-level similarities when estimating co-relevance.

Asymmetric similarity measures have been used in the vector space model [49] and the language modeling framework [20] for asymmetric co-relevance estimation. The merits of one such language model estimate [48] were demonstrated with respect to other (e.g., symmetric) language-model-based estimates [20] and (e.g., asymmetric) vector-space-based measures [49]. This estimate is integrated in, and outperformed by, our method.

Models utilizing true or pseudo-relevance feedback [23] could also be viewed as addressing a general asymmetric co-relevance estimation task. These models aim to estimate the relevance of a document given that a *few* other documents are (or are assumed to be) relevant. Such models leverage the commonalities between the given documents (e.g., shared terms). In our settings such commonalities cannot be used as a *single* document is fixed, and co-relevance is estimated with respect to this document. However, we also modify a technique developed for (pseudo-) relevance feedback, term clipping[1] [1], to produce a query-biased model of the input document here. Using the cross entropy between document models results in an asymmetric co-relevance estimate that is, to the best of our knowledge, another contribution of our work. Furthermore, this estimate often outperforms other estimates that are used for reference comparisons, although the performance is in most cases worse than that attained by using our method.

Finally, we note that in contrast to previous work on co-relevance estimation, ours is the first that applies a learning-to-rank approach and uses query-independent document-quality measures.

3. ASYMMETRIC CO-RELEVANCE

The core task addressed by probabilistic retrieval methods is to estimate the probability that a given document d is relevant to a given query q [36]. There are retrieval settings where the goal is to estimate *co-relevance*; that is, the probability that documents d and d' are both relevant to q. For example, many cluster-based document retrieval methods rank document clusters by their presumed relevance to the query [27, 16, 29, 33]. Estimating cluster relevance amounts to estimating co-relevance of documents in the cluster.

The focus of much work on probabilistic retrieval methods is estimating the relevance likelihood of a single document [36, 21]. Similarly, our starting point is estimating the likelihood of *symmetric* co-relevance:

$$p(d, d'|R = 1, q), \tag{1}$$

where R ($\in \{0, 1\}$) is a binary-relevance random variable. That is, the goal is to estimate the likelihood that d and d' are an unordered pair of documents relevant to q. Such estimation can rely, for example, not only on the similarity of each document to the query, but also on the similarity between the two documents. Equation 1 can be written

$$p(d'|d, R = 1, q)p(d|R = 1, q). \tag{2}$$

Now, suppose that we fix document d and set as a goal to rank documents d' by the resultant symmetric co-relevance likelihood of pairing them with d. In that case, following Equation 2, the goal becomes estimating the relevance likelihood of d' assuming that d is relevant:

$$p(d'|d, R = 1, q). \tag{3}$$

We refer to the resulting estimation task, which is our focus in this paper, as *asymmetric* co-relevance estimation. As already noted, quite a few retrieval methods and applications rely on such estimation (e.g., [40, 11, 7, 16, 29, 20]). For example, nearest-neighbor clustering which is often applied in cluster-based document retrieval [28, 16, 29, 33]; graph-based re-ranking methods [49, 7, 20]; and, Voorhees' nearest-neighbor cluster hypothesis test [40]. These applications are used in Section 5.2 for empirical evaluation of the estimate we devise and the reference comparisons.

4. LEARNING CO-RELEVANCE

Many methods that rely on asymmetric co-relevance estimation operate on top-retrieved documents returned by a retrieval method so as to re-rank them [28, 16, 29, 20, 33]. Therefore, we design our experimental setup to follow similar constraints, and apply our estimations only to documents in $\mathcal{D}_{\text{init}}$: an initial result list of documents from corpus $\mathcal{D}$ that are the most highly ranked in response to query q by a retrieval method; $sim_{\text{init}}(q, d)$ is the initial retrieval score assigned to document d in $\mathcal{D}_{\text{init}}$. As we show next, this allows us to view asymmetric co-relevance estimation as fusion of document rankings.

To estimate the co-relevance of documents in $\mathcal{D}_{\text{init}}$ to a fixed document d in $\mathcal{D}_{\text{init}}$ (Equation 3), we use a set of permutations (rankings) of $\mathcal{D}_{\text{init}}$, denoted $\Pi(\mathcal{D}_{\text{init}})$. A permutation π is induced using a document ranking function that assigns document d' ($\in \mathcal{D}_{\text{init}}$) the score: $score_{\pi}(d'; q, d)$. The ranking function can be based on (i) the similarity between d' and q; (ii) the similarity between d' and d which can be computed in a query dependent or independent fashion; and, (iii) the estimated quality of d' which serves for a relevance prior. We then define the co-relevance estimate of document d' with respect to d as

$$CoRel(d'; d, q) \overset{def}{=} \sum_{\pi \in \Pi(\mathcal{D}_{\text{init}})} f_{\pi}(d'; q, d) w(\pi); \tag{4}$$

$f_{\pi}(d'; q, d)$ is the feature value of d' generated from the permutation π, and $w(\pi)$ is the permutation importance

[1] Namely, representing a document using a small set of terms which are assigned the highest probability by a language model induced from the document.

weight.[2] We note that Equation 4 represents a linear fusion of permutations [39].

Having framed co-relevance estimation as a linear function of features generated from the permutations, we next describe the ranking functions that serve to induce these permutations. Then, in Section 4.3 we describe the feature values and the approach employed for learning importance weights of permutations.

4.1 Permutations

Query-based permutation. As it was created in response to q, we use $\mathcal{D}_{\text{init}}$'s ranking as one of the permutations, π_{Q}, with the score of d' being its initial score in $\mathcal{D}_{\text{init}}$:

$$score_{\text{Q}}(d'; q, d) \overset{def}{=} sim_{\text{init}}(q, d').$$

Document-based permutations. The next set of permutations are created using an inter-textual similarity estimate $sim(\cdot, \cdot)$, using a bag-of-terms representation. Section 4.2 provides the details regarding the three estimates used in our experiments. Permutation π_{D} is created by ranking documents by their similarity with d:

$$score_{\text{D}}(d'; q, d) \overset{def}{=} sim(d, d'). \quad (5)$$

Documents can be long and topically heterogeneous. Thus, inter-document similarities, as well as document-query similarities, estimated using bag-of-terms, might not reflect aspects in specific zones (passages) in the documents. This observation has motivated much work on passage-based document retrieval (e.g., [25, 15]). Thus, the following permutations are induced by utilizing passage-based information. We write $g \in d$ to indicate that passage g belongs to document d.

Permutations π_{MaxP} and π_{AvgP} are based on the maximum and average similarity between passages in d' and d:

$$score_{\text{MaxP}}(d'; q, d) \overset{def}{=} \max_{g \in d, g' \in d'} sim(g, g');$$

$$score_{\text{AvgP}}(d'; q, d) \overset{def}{=} \operatorname*{average}_{g \in d, g' \in d'} sim(g, g').$$

Using passage-based representations for short or topically coherent documents has potential drawbacks [5]. Hence, we also use the permutations π_{MaxDP}, π_{AvgDP}, π_{MaxPD} and π_{AvgPD}. These are induced using the similarity between passages in d' and d as a whole, and vice versa. Specifically,

$$score_{\text{MaxDP}}(d'; q, d) \overset{def}{=} \max_{g' \in d'} sim(d, g');$$

$$score_{\text{AvgDP}}(d'; q, d) \overset{def}{=} \operatorname*{average}_{g' \in d'} sim(d, g');$$

$$score_{\text{MaxPD}}(d'; q, d) \overset{def}{=} \max_{g \in d} sim(g, d');$$

$$score_{\text{AvgPD}}(d'; q, d) \overset{def}{=} \operatorname*{average}_{g \in d} sim(g, d').$$

Query-document-based permutations. The ranking functions we define next compare d and d' in a query-dependent manner. Let $g_{q;x}$ be the passage most similar to q in document x: $g_{q;x} \overset{def}{=} \arg\max_{g \in x} sim(q, g)$. Then,

$$score_{\text{QPP}}(d'; q, d) \overset{def}{=} sim(g_{q;d}, g_{q;d'});$$

$$score_{\text{QDP}}(d'; q, d) \overset{def}{=} sim(d, g_{q;d'});$$

$$score_{\text{QPD}}(d'; q, d) \overset{def}{=} sim(g_{q;d}, d').$$

Document-quality-based permutations. Inspired by work on Web retrieval [4], we also use several query-independent document-quality measures to create permutations. These measures serve as estimates for the document prior relevance likelihood.

The first two measures are based on the assumption that a document containing many occurrences of highly frequent terms in the corpus (e.g., stopwords) adheres to the typical use of general language. Therefore, such a document is presumably of high quality [31, 4]. Considering a stopword as any term among the 100 most frequent alphanumeric terms in the corpus [31, 4], we define: (i) $score_{\text{SW1}}(d'; q, d)$: the ratio between the number of stopwords and non-stopwords in d'; and, (ii) $score_{\text{SW2}}(d'; q, d)$: the fraction of stopwords on a stopwords list that appear in d'.

The next two measures are based on the assumption that low content repetition in a document implies to the use of rich language [31, 4], and therefore to high document quality. Accordingly, we use the inverse compression ratio of a document: $score_{\text{ICR}}(d'; q, d)$ is the ratio between the compressed (using gzip) and uncompressed document size [31]. The second quality measure is the entropy of the term distribution in a document [4]. Formally, let $p_{d'}^{Dir[0]}(w)$ be the maximum-likelihood estimate of term w with respect to document d'. (Technical details are provided in Section 4.2.) Then,

$$score_{\text{ENT}}(d'; q, d) \overset{def}{=} -\sum_{w \in d'} p_{d'}^{Dir[0]}(w) \log p_{d'}^{Dir[0]}(w).$$

4.2 Inter-textual similarity measures

Some of the permutations used by our approach are created using an inter-textual similarity measure, $sim(\cdot, \cdot)$. We use three permutations in such cases, each instantiated using one of the three similarity measures described next.

The **LM** measure utilizes language models [22]:

$$sim_{\text{LM}}(x, y) \overset{def}{=} \exp\left(-CE\left(p_x^{Dir[0]}(\cdot) \;\middle|\middle|\; p_y^{Dir[\mu]}(\cdot)\right)\right); \quad (6)$$

CE is the cross entropy measure; $p_z^{Dir[\mu]}(\cdot)$ is the Dirichlet-smoothed unigram language model induced from z with a smoothing parameter μ; here and after we set $\mu = 1000$ [22]; $p_x^{Dir[0]}(w)$ is the maximum likelihood estimate of term w with respect to x. This standard language-model measure was used in work on cluster-based and passage-based document retrieval [16, 17, 15, 20, 33].

The **Cos** measure [35], $sim_{\text{Cos}}(x, y)$, is the cosine between the tf.idf vectors representing x and y. Raw tf values are used. Herein, the idf value of term w is defined in all measures as: $idf_w \overset{def}{=} \log \frac{|\mathcal{D}| - df_w + 0.5}{df_w + 0.5}$; $|\mathcal{D}|$ is the number of documents in the corpus, and df_w is w's document frequency.

The **BM25** estimate [34], $sim_{\text{BM25}}(x, y)$, which was shown to be effective for estimating inter-document similarities [44], is the Okapi BM25 score assigned to y treating x as a query.

[2] The co-relevance estimates for documents d' ($\in \mathcal{D}_{\text{init}}$) do not constitute a probability distribution over $\mathcal{D}_{\text{init}}$, although they can be sum-normalized to that end. This normalization does not affect the ranking of documents with respect to d. We also note that the co-relevance estimate in Equation 4 is assumed to be correlated with the co-relevance probability defined in Equation 3 by the virtue of the way it is devised.

4.3 Asymmetric co-relevance estimation

To instantiate a specific co-relevance estimate using Equation 4 and the permutations described in Section 4.1, we have to define the feature value of a document per permutation and permutation importance weights.

As feature values generated from the permutations, we use $f_\pi(d'; q, d) \overset{def}{=} |\mathcal{D}_{\text{init}}| - rank(d', \pi) + 1$, where $|\mathcal{D}_{\text{init}}|$ is the number of documents in $\mathcal{D}_{\text{init}}$ and $rank(d', \pi)$ is the rank of document d' in permutation π; the rank of the highest ranked document is 1. This is motivated by work on the Borda fusion method [3], suggesting that such rank-to-score transformation is effective when combining the scores of different ranking functions.

The importance weights $w(\pi)$ can be learned using any learning-to-rank method [24]. As detailed in Section 5.3, we use SVM^{rank} [14] in our experiments. Thus, in the spirit of past work on fusing result lists (e.g., [39, 3]), we assume that the importance of a permutation created by a ranking function can be estimated based on the importance of permutations created for other (train) queries using the same function.

Given the above, the co-relevance estimate derived from Equation 4 is essentially a Weighted Borda fusion score [3]. Thus, in what follows we use **WBorda** to refer to our proposed co-relevance estimation method.[3]

Finally, we note that non-linear (in features) ranking functions could be used as an alternative to the linear function defined in Equation 4. We also experimented with using a gradient boosted regression trees method for learning permutation importance weights [9]. However, for all the applications considered for evaluation in Section 5, the resultant performance was inferior to that of using a linear method trained with SVM^{rank}. (Actual performance numbers are omitted as they convey no additional insight.)

5. EVALUATION

We study the performance of several applications when using our WBorda method to estimate asymmetric co-relevance. We compare WBorda with a large number of reference comparison estimates. Table 1 summarizes the estimates and applications. One application relies by design on a symmetric co-relevance estimate. Thus, to symmetrize WBorda and the asymmetric reference comparisons in this case, we use:

$$\frac{CoRel(d'; d, q) + CoRel(d; d', q)}{2}. \qquad (7)$$

5.1 Reference comparison estimates

There has been very little prior work on asymmetric co-relevance estimation [49, 20]. Thus, many of the reference comparisons are symmetric inter-document similarity measures. Most of the reference comparisons are query-independent; a few use passage-based information. In comparison, the WBorda method integrates query-dependent and query-independent information, as well as passage-based.

Basic estimates. Using each of LM, Cos and BM25 (described in Section 4.2) to measure inter-document similarities constitutes a reference comparison. The resultant methods are also used in WBorda to create the three π_{D} permutations (see Equation 5). LM and BM25 are asymmetric while Cos is symmetric. Some work [20] demonstrated the merits of LM as an asymmetric co-relevance estimate with respect to other asymmetric and symmetric estimates [49, 20]. A symmetric BM25 estimate, **OK**, was specifically designed for estimating inter-document similarities [45].

Information theoretic estimate. The information theoretic estimate, **IT**, is a symmetric parameter-free inter-document similarity measure. It is defined as the ratio between the amount of information shared by two documents and the amount of information contained in each [2].

Query sensitive estimates. Query sensitive similarity estimates (QSSM) are symmetric query-biased inter-document similarities [37]. These estimates use a vector representation of the terms shared by two documents, d' and d. The cosine between this vector and that of the query is used to scale $sim_{\text{Cos}}(d', d)$ in the **QSSM(M1)** measure, and is linearly interpolated with $sim_{\text{Cos}}(d', d)$, using a free-parameter α, in the **QSSM(M3)** measure.

Probabilistic estimate. The odds-based co-relevance estimate, **ProbCoR**, is a language-model-based asymmetric query-biased inter-document similarity measure proposed for symmetric co-relevance estimation [30]. The estimate is based on a linear interpolation of the similarity between d and d' and the similarity of each to the query; an interpolation parameter α is used.[4]

Optimum clustering framework estimates. The assumption of the optimum clustering framework, **OCF**, is that documents should be deemed similar if they are relevant to the same queries [10]. A symmetric inter-document similarity estimate instantiated from OCF is:

$$sim_{\text{OCF}}(d', d) \overset{def}{=} \sum_{w \in d' \cup d} sim(w, d') sim(w, d),$$

where each term w ($\in d' \cup d$) that occurs in at least one of the two documents represents a query.[5] We use here the three estimates, LM, Cos and BM25, for $sim(\cdot, \cdot)$, which results in three OCF instantiations: OCF(LM), OCF(Cos) and OCF(BM25).

Passage-based estimates. The asymmetric **AvgMaxP** estimate [32], proposed for agglomerative clustering, uses whole-document-based and passage-based inter-document similar-

[3] If we apply min-max normalization to $score_\pi(d'; q, d)$ and use the normalized score for $f_\pi(d'; q, d)$, then the co-relevance estimate amounts to a weighted CombSUM fusion score [8, 39]. The resulting performance in the applications we consider is inferior to that of using Borda's score. (Performance numbers are omitted as they convey no additional insight).

[4] $sim_{\text{ProbCoR}}(d', d) \overset{def}{=} (1 - \alpha) sim_{\text{LM1}}(d', d) + \alpha sim_{\text{LM1}}(q, d)$ $+ sim_{\text{LM1}}(q, d')$, where $sim_{\text{LM1}}(x, y) \overset{def}{=} \log \left(\frac{\mu}{\sum_{w'} tf_{w', y} + \mu} \right)$ $+ \sum_w p_x^{Dir[0]}(w) \log \left(\frac{tf_{w,y}}{\mu p_{\mathcal{D}}^{Dir[0]}(w)} + 1 \right)$; $tf_{w,x}$ is the number of times w appears in x; $\mu = 1000$.

[5] For the large corpora we use, having each term in the vocabulary represent a query, as originally proposed [10], is computationally expensive. Therefore, we only consider terms that occur in at least one of the two documents.

ities as is the case for WBorda:

$$sim_{\text{AvgMaxP}}(d', d) \overset{def}{=} (1-\alpha)sim(d', d) + \frac{\alpha}{|\{g' : g' \in d'\}|} \sum_{g' \in d'} \max_{g \in d} sim(g', g);$$

$|\{g' : g' \in d'\}|$ is the number of passages in d'. While AvgMaxP was originally instantiated using Cos [32], here we also use LM and BM25 which results in three measures: AvgMaxP(LM), AvgMaxP(Cos) and AvgMaxP(BM25).[6]

Relevance-model-based estimate. We study an additional asymmetric co-relevance estimate, **RM3**, that utilizes relevance models [23, 1]. To the best of our knowledge, this estimate has not been utilized in the various applications that we use for evaluation. (See Section 5.2.) The co-relevance of d' to d is estimated by the similarity of d' with a query-biased representation of the terms most "dominant" in d:

$$sim_{\text{RM3}}(d', d) \overset{def}{=} \exp\left(-CE\left(p_{R(d,q)}(\cdot) \;\Big|\Big|\; p_{d'}^{Dir[\mu]}(\cdot)\right)\right);$$

$R(d, q)$ is relevance model #3 [1] which uses a parameter α:

$$p_{R(d,q)}(w) \overset{def}{=} (1-\alpha)p_q^{Dir[0]}(w) + \alpha p_{d^{clip}}^{Dir[\mu]}(w);$$

$p_{d^{clip}}^{Dir[\mu]}(\cdot)$ is a clipped language model of d attained by setting to zero the probabilities of all but the β terms (β is a free parameter) to which the Dirichlet-smoothed language model induced from d assigns the highest probability; sum-normalization is applied to the probabilities of these terms to yield a valid language model.[7]

Unweighted Borda estimate. To study the merits of using a supervised learning model to set the permutation importance weights in WBorda, we also use an unsupervised model that utilizes the same permutations but with uniform importance weights. The resultant Unweighted Borda fusion method, **UBorda**, sets $CoRel(d'; d, q) \overset{def}{=} \sum_{\pi \in \Pi(\mathcal{D}_{\text{init}})} f_\pi(d'; q, d)$. (See Equation 4.)

5.2 Applications

We next describe the applications in which WBorda, and the reference comparisons, are used. All applications, except for one (Regularization), rely by design on an asymmetric co-relevance estimate. Yet, in past work, symmetric inter-document similarities were also used in these applications. Table 1 provides a summary of the applications.

The nearest-neighbor cluster hypothesis test. Voorhees' nearest-neighbor (NN) test [40] is the most commonly used test for the cluster hypothesis. The test is also often used to compare inter-document similarities [37, 30, 45].

For each *relevant* document $d \in \mathcal{D}_{\text{init}}$, a ranking of all the documents $d' \in \mathcal{D}_{\text{init}} \setminus \{d\}$ is created using a co-relevance estimate. This is a natural asymmetric co-relevance estimation task. Ranking effectiveness is measured using precision at top ranks or average precision. The final NN test score is the average over all queries in a test set of the average ranking effectiveness for the relevant documents.

Table 1: Summary of the reference comparison estimates and applications considered.

Reference Comparisons		Applications	
Asymmetric	Symmetric	Asymmetric	Symmetric
LM	Cos	NN test	Regularization
BM25	OK	Interpf	
ProbCoR	IT	ClustMRF	
AvgMaxP	QSSM	ClustRanker	
RM3	OCF	RWI	

We note that the NN cluster hypothesis test is essentially a relevance-feedback-based retrieval task. That is, a single relevant document, $d \in \mathcal{D}_{\text{init}}$, is provided, and the task is to re-rank $\mathcal{D}_{\text{init}}$ using this relevance feedback.

Cluster-based document retrieval. There are various document re-ranking methods that utilize information induced from clusters of documents in an initially retrieved list [46, 26, 19, 28, 27, 47, 16, 29, 17, 33]. Most of these methods were applied using nearest-neighbor (NN) clustering which was shown to be highly effective with respect to other clustering schemes [17, 33].

A cluster is created for each document $d \in \mathcal{D}_{\text{init}}$ by ranking the documents $d' \in \mathcal{D}_{\text{init}} \setminus \{d\}$ with respect to d; often, inter-document similarities are used. A cluster then comprises d and the $k-1$ most highly ranked documents, i.e., d's $k-1$ nearest neighbors. Thus, $|\mathcal{D}_{\text{init}}|$ (overlapping) clusters are created, each contains k documents. Because the documents in $\mathcal{D}_{\text{init}}$ are ranked with respect to a fixed d, in a query context (i.e., $\mathcal{D}_{\text{init}}$), the task of creating a nearest-neighbor cluster amounts to estimating asymmetric co-relevance as was the case for the nearest-neighbor cluster hypothesis test. Indeed, using the precision-at-top-ranks evaluation metric in the nearest-neighbor test corresponds to measuring the percentage of relevant documents in a nearest-neighbor cluster constructed from a relevant document.

The cluster-based re-ranking methods that we use apply various approaches of utilizing cluster-based information. Thus, to ameliorate metric divergence effects [24], we evaluate the quality of a co-relevance estimate used to create the clusters with respect to re-ranking effectiveness.

The interpolation-f [17] method, **Interpf**, interpolates the (normalized) initial retrieval score of a document, $sim_{\text{init}}(q, d)$, with its (normalized) cluster-based score; λ is a free interpolation parameter. Interpf could be viewed as a generalized version of methods that apply cluster-based [26] and topic-based [43] document language model smoothing.

Interpf ranks documents directly using cluster-based information. Some other cluster-based methods apply a two-steps procedure. First, document clusters are ranked based on their presumed relevance to the information need underlying the query. Then, the cluster ranking is transformed to document ranking by replacing each cluster with its documents while omitting repeats; the order of documents in a cluster is determined by their initial retrieval scores. The two state-of-the-art cluster-based methods that we consider differ in the way by which clusters are ranked with respect to the query, i.e., the rankings produced in the first step.

The **ClustMRF** method ranks a cluster based on integrating, using a learning-to-rank approach, different information types that presumably attest to the cluster relevance [33]. Some of these information types, the stopwords-based measures, SW1, SW2, the inverse compression ratio

[6] To avoid using negative values in the estimates, min-max normalization was applied to the similarity values of BM25, OK, ProbCoR, OCF(BM25) and AvgMaxP(BM25).

[7] Using RM3 in our WBorda method to produce a permutation did not result in consistent, or substantial, performance improvements in the applications we consider.

ICR, and the entropy of the term distribution ENT, are also used by WBorda to create permutations. We use WBorda, and the reference comparisons, not only to create the clusters, but also to induce the inter-document similarities used for measuring cluster cohesion in ClustMRF.

The **ClustRanker** method [16] ranks clusters using cluster-query, document-query, inter-document and inter-cluster similarities. We use WBorda, and the reference comparisons, in ClustRanker not only to induce clusters, but also to compute inter-document similarities for inducing document centrality. A free parameter λ controls the balance between using whole-cluster-based and document-based information.

In all the cluster-based retrieval methods described above, query-cluster, document-cluster and inter-cluster similarities are induced using the LM estimate from Equation 6.[8]

Graph-based document retrieval. The graph-based re-ranking methods that we consider use a co-relevance estimate to determine the nearest neighbors of a document and to set edge weights in a weighted nearest-neighbor graph. We evaluate the quality of the co-relevance estimate used with respect to the resultant re-ranking effectiveness.

The recursive weighted influx method (**RWI**) [20] induces document centrality over a nearest-neighbor graph composed of the documents in $\mathcal{D}_{\text{init}}$; the PageRank algorithm is used with a dumping factor γ. Documents in $\mathcal{D}_{\text{init}}$ are re-ranked by scaling their initial retrieval score, $sim_{\text{init}}(q, d)$, with their centrality value. The weight of a directed edge connecting document d with one of its nearest neighbors d' reflects the likelihood that d' is relevant assuming that d is relevant. Thus, to determine the nearest neighbors of a document, and to set edge weights, asymmetric co-relevance estimates are used. Originally, an asymmetric language model inter-document similarity measure was used.[9]

Following the cluster hypothesis, the **Regularization** re-ranking method is based on the premise that similar documents in $\mathcal{D}_{\text{init}}$ should be assigned with similar retrieval scores [7]. The method uses a nearest-neighbor graph to regularize (with a parameter γ) the initial retrieval score of document d, $sim_{\text{init}}(q, d)$, using the scores of its nearest neighbors. A *symmetric* co-relevance estimate is used to determine nearest neighbors and to set edge weights. Originally, the multinomial diffusion kernel, **MDK**, which relies on a free-parameter t, was used.

5.3 Experimental setup

Data & evaluation metrics. We conducted our experiments on several TREC datasets specified in Table 2. AP and ROBUST are small collections, composed mainly of news articles. WT10G is a small Web collection and GOV2 is a crawl of the .gov domain. CW09B is the Category B of the ClueWeb09 Web collection and CW12B is the Category B of ClueWeb12. Two additional Web settings, CW09BF and CW12BF, were created by filtering out from the initial rankings of CW09B and CW12B, respectively, documents assigned with a score below 50 by Waterloo's spam classifier [6]. The residual corpus rankings were used to create the initial result lists, which presumably contain fewer spam documents than those for CW09B and CW12B.

Table 2: Data used for experiments.

corpus	# of documents	data	queries
AP	242,918	Disks 1-3	51-150
ROBUST	528,155	Disks 4-5 (-CR)	301-450,600-700
WT10G	1,692,096	WT10g	451-550
GOV2	25,205,179	GOV2	701-850
CW09B	50,220,423	ClueWeb09 Category B	1-200
CW09BF			
CW12B	52,343,021	ClueWeb12 Category B	201-250
CW12BF			

Topic titles served as queries. We applied Krovetz stemming to queries and documents, and removed stopwords from queries using the INQUERY list. The Indri toolkit[10] was used for experiments.

All the applications described in Section 5.2 operate on an initial result list, $\mathcal{D}_{\text{init}}$. The list contains the $|\mathcal{D}_{\text{init}}| = 50$ documents d in the corpus that yield the highest $sim_{\text{init}}(q, d) \overset{def}{=} sim_{\text{LM}}(q, d)$; i.e., a standard language-model-based initial ranking is used. Such a short (often, language-model-based) initial result list was also used in the original reports on the applications we use here [40, 7, 16, 17, 33].

The mean average precision of the 50 documents in $\mathcal{D}_{\text{init}}$ (**MAP**) and the precision of the top-5 documents (**p@5**) serve for retrieval evaluation metrics. Statistically significant differences of performance are determined using the two-tailed paired t-test with a 95% confidence level.

Training parameters. The permutation importance weights used in WBorda were learned using SVM^{rank} [14] applied with a linear kernel and default free-parameter values. In the learning phase, each document d ($\in \mathcal{D}_{\text{init}}$) which is relevant to q served as a "pseudo query" for which all other documents d' ($\in \mathcal{D}_{\text{init}} \setminus \{d\}$) are either relevant or not according to TREC's binary relevance judgements for q.[11]

The permutation weights, the free-parameter values of the reference comparison estimates, and those of the retrieval methods that serve as applications, were set using ten-fold cross validation; folds were created based on query IDs. Unless stated otherwise, MAP served as the optimization criterion in the learning phase. For WBorda, the same training set was used to learn the permutation weights and to set the passage size. We used half-overlapping fixed-size windows of terms for passages as these were shown to be highly effective for passage-based retrieval [25, 15]. The window size used for WBorda and AvgMaxP was selected from $\{100, 150, 200, 250, 300\}$.

The values of the parameters α (QSSM(M3), ProbCoR, AvgMaxP and RM3) and λ (Interpf, ClustRanker and ClustMRF) were selected from $\{0, 0.1, \ldots, 1\}$. The number of nearest neighbors, k, used in RWI and Regularization, is in $\{5, 10, 25, 50\}$. The size, k, of the nearest-neighbor clusters used, specifically, in Interpf, ClustRanker and ClustMRF, was set to 5 following the original reports [16, 17, 33]. The values of t^{-1} (Regularization) and γ (RWI and Regulariza-

[8] A cluster is represented by the big document that results from concatenating its constituent documents [28, 16, 29, 17]. The order of concatenation has no effect since unigram language models are used.

[9] The measure [20] is closely connected to the LM measure (Equation 6) in that it uses the exponent of the negative KL divergence between the documents' language models.

[10] `www.lemurproject.org/indri`

[11] Learning the weights by ranking documents with respect to both relevant and non-relevant documents resulted in less effective performance.

Table 3: Voorhees' nearest-neighbor cluster hypothesis test. 'b' marks statistically significant difference with WBorda. The best result in a column is boldfaced.

	AP		ROBUST		WT10G		GOV2		CW09B		CW09BF		CW12B		CW12BF	
	MAP	p@4	MAP	p@4	MAP	p@4	MAP	p@4	MAP	p@4	MAP	p@4	MAP	p@4	MAP	p@4
LM	12.8	77.0	21.6$_b$	68.5$_b$	13.5$_b$	62.5$_b$	14.2$_b$	81.3$_b$	23.2$_b$	72.5$_b$	23.6$_b$	71.9$_b$	24.5$_b$	51.4$_b$	20.9$_b$	47.6$_b$
Cos	11.4$_b$	70.4$_b$	19.6$_b$	62.2$_b$	13.6$_b$	62.4$_b$	14.2$_b$	80.1$_b$	22.7$_b$	69.1$_b$	23.4$_b$	70.6$_b$	25.3$_b$	54.8$_b$	21.6$_b$	51.8$_b$
BM25	12.8	77.1	22.0$_b$	68.8$_b$	14.3$_b$	63.1$_b$	14.6$_b$	83.2$_b$	23.0$_b$	71.4$_b$	23.5$_b$	71.3$_b$	24.6$_b$	53.1$_b$	21.2$_b$	48.5$_b$
IT	12.8	77.0	21.8$_b$	68.9$_b$	14.2$_b$	64.4$_b$	14.5$_b$	81.8$_b$	23.5	73.0	23.8$_b$	71.9$_b$	26.0$_b$	54.7	22.6	53.7
OK	12.8	**78.1**	22.1$_b$	69.8$_b$	14.4$_b$	64.0$_b$	14.7$_b$	83.6$_b$	23.0$_b$	71.0$_b$	23.7$_b$	71.1$_b$	25.1$_b$	53.9$_b$	21.4$_b$	48.2$_b$
QSSM(M1)	11.6$_b$	66.8$_b$	20.1$_b$	59.0$_b$	13.0$_b$	58.4$_b$	13.5$_b$	76.2$_b$	20.8$_b$	58.9$_b$	22.6$_b$	66.9$_b$	23.0$_b$	49.7$_b$	19.4$_b$	44.1$_b$
QSSM(M3)	11.8$_b$	70.6$_b$	20.7$_b$	64.9$_b$	13.8$_b$	63.5$_b$	14.2$_b$	80.1$_b$	22.7$_b$	69.1$_b$	23.4$_b$	70.3$_b$	25.2$_b$	53.9$_b$	21.4$_b$	51.8$_b$
ProbCoR	**13.0**	77.9	23.5	73.6	14.2$_b$	64.2$_b$	14.4$_b$	82.9$_b$	23.4$_b$	72.2$_b$	24.3$_b$	74.3$_b$	25.3$_b$	54.2	22.0	50.2$_b$
OCF(LM)	10.6$_b$	60.0$_b$	16.1$_b$	42.8$_b$	11.2$_b$	52.7$_b$	13.2$_b$	71.3$_b$	19.7$_b$	47.2$_b$	21.6$_b$	60.7$_b$	23.8$_b$	53.0$_b$	20.3$_b$	47.9$_b$
OCF(Cos)	11.4$_b$	70.4$_b$	19.6$_b$	62.2$_b$	13.6$_b$	62.4$_b$	14.2$_b$	80.1$_b$	22.7$_b$	69.1$_b$	23.4$_b$	70.6$_b$	25.3$_b$	54.8$_b$	21.6$_b$	51.8$_b$
OCF(BM25)	12.8	77.7	22.5$_b$	70.5$_b$	14.3$_b$	63.6$_b$	14.6$_b$	83.4$_b$	22.9$_b$	70.7$_b$	23.7$_b$	72.3$_b$	25.0$_b$	51.3$_b$	21.5	48.9$_b$
AvgMaxP(LM)	12.9	77.6	22.6$_b$	70.5$_b$	14.7$_b$	65.3$_b$	14.6$_b$	81.8$_b$	23.3$_b$	72.4$_b$	23.8$_b$	71.5$_b$	25.9$_b$	55.9	22.1	48.4$_b$
AvgMaxP(Cos)	11.4$_b$	70.7$_b$	19.7$_b$	62.1$_b$	13.9$_b$	64.3$_b$	14.3$_b$	80.5$_b$	23.0$_b$	71.0$_b$	23.6$_b$	71.7$_b$	26.4	58.8	22.0	51.7$_b$
AvgMaxP(BM25)	12.8	77.2	22.0$_b$	69.1$_b$	14.2$_b$	63.8$_b$	14.7$_b$	83.3$_b$	22.9$_b$	70.8$_b$	23.6$_b$	72.3$_b$	25.4$_b$	53.3$_b$	21.3$_b$	48.9$_b$
RM3	12.9	77.0	23.4$_b$	73.2$_b$	14.6$_b$	67.5$_b$	14.7$_b$	82.9$_b$	**23.7**	73.4	24.5	73.8$_b$	26.5	56.4	23.2	52.8$_b$
UBorda	12.7	76.6	22.8$_b$	71.3$_b$	**15.5**	69.7	14.8$_b$	82.6$_b$	23.6	72.4$_b$	24.2$_b$	72.3$_b$	26.9	57.2	23.2	55.1
WBorda	12.9	76.9	**23.7**	**74.4**	15.4	**71.0**	**15.1**	**85.1**	**23.7**	**74.8**	**24.7**	**76.9**	**28.3**	**62.4**	**24.6**	**62.2**

tion) were selected from $\{0.1, 0.2, \ldots, 0.9\}$. The number of terms, β, used in RM3 was set to a value in $\{25, 50, 75, 100\}$. The free parameters of Okapi BM25, k_1 and b, are set as follows. When using the BM25 inter-textual estimate in our WBorda method, and in the OCF(BM25) and AvgMaxP(BM25) reference comparisons, default parameter values are used: $k_1 = 1.2$ and $b = 0.75$. When using BM25 as a stand-alone reference comparison, or in the highly effective recently proposed OK estimate that serves as a reference comparison [45], k_1 and b are set to values in $\{1.2, 2, 4, 8, 12\}$, and $\{0.25, 0.5, 0.75, 1\}$, respectively.

5.4 Experimental results

In this section, we compare the effectiveness of WBorda with that of the various reference comparison estimates in the different applications (summarized in Section 5.2).

The nearest-neighbor cluster hypothesis test. The results of the nearest-neighbor (NN) cluster hypothesis test, applied with the various co-relevance estimates, are presented in Table 3. To evaluate the effectiveness of the rankings created with respect to each relevant document, we used MAP, computed for the 49 documents in $\mathcal{D}_{\text{init}} \setminus \{d\}$, and p@4, computed for the 4 most highly ranked documents.[12] The free-parameter values were set to optimize MAP and p@4 separately (on the train query set), as each of the two metrics is a somewhat different quantification of the extent to which the cluster hypothesis holds according to the test.

Our main observation based on Table 3 is that in the vast majority of the cases the cluster hypothesis, as measured by the NN test, holds to the largest extent when the WBorda estimate is used. Although WBorda is outperformed in some cases by some reference comparisons for AP and by UBorda for WT10G, the differences are not statistically significant.

In a pairwise comparison of RM3 with each of the other estimates, we see that in most relevant comparisons (8 experimental settings × 2 evaluation metrics) the test performance attained for RM3 is better; we note that many of these differences are also statistically significant. (To avoid cluttering the table, we do not mark the significance of differences between the reference comparison estimates.) Recall that using RM3 as a co-relevance estimate for the NN cluster hypothesis test, as well as for the retrieval methods we consider below, is novel to this study. Nevertheless, the test performance for RM3 is almost always lower — often to a statistically significant degree — than that for WBorda. The only case in which RM3 outperforms WBorda is p@4 for AP, yet the difference is not statistically significant.

Comparing the weighted WBorda estimate with its unweighted version UBorda, we see that the test performance for UBorda is better than that for WBorda only in a single case of MAP for WT10G, but the difference is not statistically significant. This finding attests to the merits of using a supervised model to set the permutation weights in WBorda.

Table 4: Top four permutations. The inter-textual similarity estimates used to instantiate the permutations are specified in parentheses.

	1	2	3	4
AP	D(BM25)	Q	AvgDP(LM)	AvgP(LM)
ROBUST	Q	ICR	MaxDP(LM)	MaxPD(BM25)
WT10G	SW2	QPP(Cos)	ICR	MaxP(BM25)
GOV2	AvgPD(Cos)	Q	AvgP(LM)	ICR
CW09B	SW1	Q	MaxDP(Cos)	AvgDP(BM25)
CW09BF	Q	AvgPD(Cos)	SW2	AvgPD(BM25)
CW12B	AvgPD(Cos)	SW1	AvgP(Cos)	ENT
CW12BF	Q	AvgPD(Cos)	ENT	SW1

Permutation analysis. We next turn to study the relative importance of the (35) permutations used by WBorda. The permutation weights are learned in an application-independent manner. That is, a relevant document is fixed and all other documents are ranked with respect to this document and the query. (Refer to Section 5.3 for details.) The passage size used by WBorda is learned per application. Thus, to analyze permutation importance in an application-independent way, we averaged over folds and passage sizes the weights assigned to the permutations by SVM^{rank} in the learning phase. Table 4 presents for each experimental setting the four permutations assigned with the highest averaged weights.

We can see that all four types of permutations (query-based, document-based, query-document-based and document-quality-based) have at least one representative in Table 4.

[12] We use p@4 since using four nearest neighbors of a document conceptually corresponds to using nearest-neighbor clusters of five documents as is the case for the cluster-based retrieval methods we evaluate below.

Table 5: Using Interpf, ClustMRF and ClustRanker to re-rank the initial list. '*i*' and '*b*' mark statistically significant differences with the initial ranking (Init) and WBorda, respectively. The best result in a column for a cluster-based method is boldfaced.

	AP		ROBUST		WT10G		GOV2		CW09B		CW09BF		CW12B		CW12BF	
	MAP	p@5	MAP	p@5	MAP	p@5	MAP	p@5	MAP	p@5	MAP	p@5	MAP	p@5	MAP	p@5
Init	9.0	43.6	18.8	48.7	13.8	33.4	11.4	55.5	9.9	22.7	12.7	34.7	14.2	23.6	14.0	22.8
Interpf																
LM	9.1	46.1	19.4^i	48.8_b	13.4	32.6_b	12.0^i_b	56.4_b	12.2^i_b	32.7^i_b	13.5^i_b	38.0_b	14.2	23.6	13.6	22.8
Cos	8.8_b	43.4	19.2	49.2	13.8	35.7	11.9^i_b	55.0_b	11.8^i_b	29.4^i_b	13.5^i_b	37.5_b	14.2	23.6	13.3	21.2
BM25	9.1	**48.3**	19.4^i	48.4_b	13.7	33.8_b	12.0^i_b	57.2_b	12.1^i_b	31.4^i_b	13.3^i_b	37.0_b	14.0	22.8	**14.3**	22.4
IT	9.2	46.9	$\mathbf{19.7}^i$	49.2	**13.9**	35.5	12.0^i_b	56.1_b	12.3^i_b	31.7^i_b	13.6^i_b	37.0_b	14.2	23.6	13.7	23.6
RM3	9.2	46.1	19.5^i	49.9	12.9	34.4	12.1^i_b	58.2	12.6^i	35.1^i	13.9^i	38.4^i_b	14.2	20.8	13.8	22.0
UBorda	9.2	46.9	19.4^i	49.0_b	13.0	35.3	12.0^i_b	56.4_b	12.0^i_b	31.2^i_b	13.4^i_b	36.4_b	14.2	23.6	13.6	20.8
WBorda	**9.3**	47.7	19.4^i	**50.4**	13.3	**36.5**	$\mathbf{12.6}^i$	$\mathbf{61.5}^i$	$\mathbf{12.8}^i$	$\mathbf{36.4}^i$	$\mathbf{14.4}^i$	$\mathbf{41.1}^i$	**16.7**	**25.2**	13.9	**24.0**
ClustMRF																
LM	9.3	42.2	19.4	48.6_b	14.3	36.7	$\mathbf{12.7}^i$	$\mathbf{66.1}^i$	13.7^i	$\mathbf{40.3}^i$	$\mathbf{14.8}^i$	41.4^i	20.9^i	32.0^i	18.5^i	33.2^i
Cos	8.4^i_b	40.2	18.4_b	48.6_b	14.8	37.7	12.4^i	62.8^i	13.6^i	37.5^i	14.2^i	39.9^i	21.1^i_b	35.2^i	19.5^i	30.8^i
BM25	9.1	**43.8**	19.3	51.1	13.8	37.9	12.2^i_b	64.1^i	13.3^i_b	38.4^i	14.7^i	42.6^i	18.7^i	32.0^i	16.9	29.2
OCF(BM25)	9.5	42.8	$\mathbf{19.6}^i$	50.9	13.9	39.0	12.0^i_b	61.8^i	12.8^i_b	37.5^i	14.4^i	$\mathbf{43.4}^i$	21.2^i	31.2	$\mathbf{21.7}^i$	32.0^i
AvgMaxP(LM)	9.0_b	42.0	19.5^i	50.8	14.7	38.1	12.6^i	65.1^i	13.7^i	38.1^i	14.6^i	41.6^i	$\mathbf{21.9}^i$	32.0^i	18.4^i	30.0
UBorda	9.2_b	**43.8**	$\mathbf{19.6}^i$	50.8	$\mathbf{15.8}^i$	$\mathbf{40.6}^i$	12.4^i_b	63.0^i	13.2^i_b	39.1^i	14.5^i	41.2^i	21.4^i	$\mathbf{35.6}^i$	20.3^i	34.4^i
WBorda	**9.6**	43.6	$\mathbf{19.6}^i$	$\mathbf{52.1}^i$	14.9	37.9	$\mathbf{12.7}^i$	65.7^i	$\mathbf{14.0}^i$	40.2^i	14.6^i	42.8^i	19.0^i	34.8^i	20.5^i	$\mathbf{35.2}^i$
ClustRanker																
LM	9.3	47.5	19.2	46.2_b	13.2	29.9	12.1^i_b	58.0_b	12.2^i_b	32.2^i_b	12.8_b	32.3_b	13.1	20.8_b	10.2^i_b	13.2^i_b
Cos	8.7	41.8	17.9^i_b	42.8^i_b	12.4_b	28.9	11.6_b	53.8_b	11.5^i_b	28.6^i_b	12.9_b	32.0_b	12.6_b	17.6^i_b	13.0_b	21.6_b
BM25	9.0	45.9	**19.3**	47.4_b	12.7	30.3	12.3^i	58.5_b	11.9^i_b	29.9^i_b	13.0_b	35.7_b	16.4	23.2	12.8_b	16.8_b
ProbCoR	**9.4**	**47.7**	19.1	45.9_b	13.3	32.8	11.8_b	53.5_b	12.6^i	33.4^i_b	13.1_b	35.6_b	14.0	22.4	12.5_b	21.2_b
RM3	8.9	45.7	18.3_b	43.8^i_b	13.0	29.7	12.2^i	58.5_b	12.4^i_b	32.7^i_b	$\mathbf{14.1}^i$	39.8^i	$\mathbf{18.5}^i_b$	$\mathbf{31.6}^i$	12.9_b	23.6_b
UBorda	8.5_b	40.4_b	19.1	48.7	13.7	**33.2**	12.4^i	61.8^i	11.6^i_b	29.4^i_b	$\mathbf{14.1}^i$	39.4^i	14.2	20.8_b	14.5	24.8_b
WBorda	9.0	45.7	19.1	**50.4**	**13.8**	32.2	$\mathbf{12.6}^i$	$\mathbf{65.1}^i$	$\mathbf{13.1}^i$	$\mathbf{38.6}^i$	14.0^i	$\mathbf{40.5}^i$	15.9	29.6	$\mathbf{19.3}^i$	$\mathbf{34.4}^i$

The query-based permutation π_{Q} is almost always among the top four permutations[13], which attests to the merits of using the query for estimating asymmetric co-relevance. One of the document-based permutations that uses passages, i.e., π_{MaxP}, π_{AvgP}, π_{MaxDP}, or π_{AvgPD} is always among the top four. This finding attests to the merit in using passage-based information to estimate asymmetric co-relevance. The query-document-based permutations have a single occurrence in Table 4 (QPP(Cos)); however, they are always among the top nine permutations. The AP setting is the only one for which a document-quality-based permutation, π_{SW1}, π_{SW2}, π_{ICR} or π_{ENT}, is not among the top four permutations. It is also the only setting for which WBorda was outperformed when using a reference comparison estimate (excluding UBorda) for the nearest-neighbor cluster hypothesis test reported in Table 3.

Comparing the results in Table 3 and Table 4, we see that when used alone for estimating the extent to which the cluster hypothesis holds, the performance attained for Cos is often lower than that attained for LM and BM25. Conversely, permutations instantiated using Cos have more representatives in the top four permutations than those instantiated using BM25 and LM.

Retrieval performance. In what follows, we study the effectiveness of the retrieval methods discussed in Section 5.2 when using the co-relevance estimates. The estimates are used to determine the nearest neighbors of a document in the graph-based methods and set the edge weights, create the NN clusters in the cluster-based methods, and induce some of the other features used by the methods. As reference comparisons to WBorda, we use the LM, Cos and BM25 estimates which it integrates. For each retrieval method we present the performance of two additional reference comparison estimates which led to the best performance in most relevant comparisons compared to the other reference comparisons. We also present the performance of the unweighted UBorda estimate. We note that some of the performance numbers reported here are not comparable with those reported in past literature for the following reasons: (i) we use larger sets of queries than those used in past work addressing the same datasets (e.g., [4, 33]); and (ii) we report MAP@50, as our focus is on re-ranking an initial list of 50 documents, rather than MAP@1000 (e.g., [7, 4]).[14]

Cluster-based methods. Table 5 presents the results for the cluster-based re-ranking methods. We can see that using WBorda yields the best performance in most relevant comparisons (8 experimental settings × 2 evaluation metrics) for Interpf and ClustRanker. For both methods, WBorda outperforms any reference comparison in a vast majority of the relevant comparisons with many of the improvements being statistically significant. In the few cases where WBorda is outperformed by a reference comparison, the performance differences are statistically indistinguishable.

We can also see that WBorda outperforms any reference comparison in a vast majority of the relevant comparisons for ClustMRF; many of the improvements are statistically significant. In addition, WBorda is statistically significantly outperformed only in a single case for the CW12B setting.

The findings presented above attest to the clear merits of using WBorda to induce nearest-neighbor clusters that are used by highly effective cluster-based retrieval methods.

[13] For WT10G π_{Q} is among the top six permutations and for CW12B it is among the top five.

[14] The p@5 performance for cluster-based retrieval methods might be somewhat lower than that reported in past literature as the optimization metric here was MAP.

Table 6: Using the graph-based RWI and Regularization methods to re-rank the initial list. 'i' and 'b' mark statistically significant differences with the initial ranking (Init) and WBorda, respectively. The best result in a column per a graph-based method is boldfaced.

	AP		ROBUST		WT10G		GOV2		CW09B		CW09BF		CW12B		CW12BF	
	MAP	p@5	MAP	p@5	MAP	p@5	MAP	p@5	MAP	p@5	MAP	p@5	MAP	p@5	MAP	p@5
Init	9.0	43.6	18.8	48.7	13.8	33.4	11.4	55.5	9.9	22.7	12.7	34.7	14.2	23.6	14.0	22.8
RWI																
LM	8.9_b	46.1	19.2_b	49.9_b	12.8_b	35.1_b	12.0^i_b	60.5^i_b	12.2^i_b	33.1^i_b	13.2_b	38.0_b	14.3_b	23.6_b	16.2_b	25.2_b
Cos	9.0	44.2_b	18.9_b	48.8_b	13.8	37.1^i_b	12.0^i_b	60.8^i_b	11.6^i_b	31.6^i_b	13.6^i_b	37.4_b	15.3^i_b	24.4_b	14.7_b	25.2_b
BM25	9.0	47.9	18.9_b	49.4_b	13.8	35.5_b	12.7^i_b	63.1^i	11.6^i_b	30.8^i_b	13.5^i_b	39.8^i_b	14.2_b	22.0_b	13.9_b	24.8_b
IT	8.9_b	46.1_b	19.0_b	49.2_b	13.1_b	35.9_b	12.6^i_b	61.6^i_b	12.5^i_b	34.8^i_b	14.1^i	38.9_b	14.6_b	24.8_b	16.4_b	25.2_b
RM3	8.9_b	44.8_b	19.3	48.7_b	13.8	36.5_b	12.7^i_b	62.0^i_b	12.5^i_b	33.7^i_b	14.0^i	40.3^i_b	14.8_b	27.6	15.6_b	28.4_b
UBorda	9.0	46.7	19.1_b	50.0_b	**14.2**	37.5^i_b	12.8^i_b	64.2^i	12.2^i_b	32.4^i_b	14.2^i	41.2^i_b	14.9_b	25.6_b	15.5_b	27.6^i_b
WBorda	**9.3**	$\mathbf{49.1}^i$	$\mathbf{19.7}^i$	$\mathbf{52.5}^i$	**14.2**	$\mathbf{40.4}^i$	$\mathbf{13.1}^i$	$\mathbf{65.9}^i$	$\mathbf{13.1}^i$	$\mathbf{39.5}^i$	$\mathbf{14.6}^i$	$\mathbf{43.8}^i$	$\mathbf{18.4}^i$	$\mathbf{33.2}^i$	$\mathbf{18.7}^i$	$\mathbf{34.0}^i$
Regularization																
MDK	9.0	44.2_b	19.7^i_b	50.1_b	13.9	35.5	12.2^i_b	59.1^i_b	12.3^i	32.7^i	13.4^i_b	37.3_b	13.9_b	24.0_b	14.3_b	24.0_b
Cos	8.8_b	44.2_b	19.2^i_b	48.8_b	13.6_b	37.1	11.9^i_b	58.8_b	11.6^i_b	28.7^i_b	13.6^i_b	39.2^i	13.8_b	24.0_b	14.8_b	26.4^i
BM25	9.1	46.5	20.2^i	51.0^i	14.3	36.5	12.1^i_b	58.1_b	12.4^i	$\mathbf{34.8}^i$	13.7^i_b	38.2_b	13.0_b	20.8_b	15.9	21.6_b
OCF(BM25)	9.2	46.1	20.1^i	50.2_b	14.3^i	35.3_b	12.3^i_b	58.5_b	$\mathbf{12.7}^i$	34.6^i	13.8^i	39.3^i	13.8_b	23.2_b	15.7	21.6_b
RM3	9.0_b	45.1_b	19.7^i_b	49.9_b	**15.1**	38.8^i	12.6^i_b	61.2^i	12.0^i	30.1^i_b	$\mathbf{14.4}^i$	41.1^i	15.5_b	26.4_b	14.9	27.6
UBorda	$\mathbf{9.3}^i$	46.9	20.2^i	51.7^i	14.9	$\mathbf{40.4}^i$	12.6^i_b	62.6^i	11.8^i_b	31.4^i_b	13.9^i_b	39.8^i	15.1_b	24.4_b	14.1_b	24.4_b
WBorda	$\mathbf{9.3}^i$	$\mathbf{48.1}^i$	$\mathbf{20.3}^i$	$\mathbf{52.4}^i$	14.8	39.8^i	$\mathbf{12.9}^i$	$\mathbf{64.3}^i$	12.2^i	33.2^i	14.3^i	$\mathbf{41.6}^i$	$\mathbf{18.4}^i$	$\mathbf{30.4}^i$	$\mathbf{17.3}^i$	$\mathbf{28.8}^i$

Graph-based methods. Table 6 presents the results for the graph-based RWI and Regularization methods. We see that for both RWI and Regularization the best performance is almost always attained when WBorda is used. Most of the improvements over the reference comparisons are substantial and statistically significant. Furthermore, in the few cases where WBorda is outperformed by one of the reference comparisons, the performance differences are not statistically significant.

We make the following additional observations. The RWI method relies on an asymmetric co-relevance estimate. Our asymmetric WBorda estimate leads to better performance of RWI than all symmetric and asymmetric reference comparisons, including the originally proposed asymmetric LM estimate [20]. On the other hand, Regularization relies on a symmetric co-relevance estimate. The symmetric version of WBorda used in Regularization yields better performance than all the symmetric reference comparisons, namely MDK (which is the originally proposed measure for Regularization [7]), Cos and OCF(BM25) and all the symmetric versions of the asymmetric reference comparison estimates: BM25, RM3 and UBorda.

Finally, we note that for both the cluster-based and graph-based re-ranking methods, WBorda outperforms UBorda in a vast majority of the relevant comparisons with quite a few of the improvements being statistically significant. This finding, which is in line with those reported above for the cluster hypothesis test, attests to the merits of learning permutation weights rather than using uniform weights.

Symmetric vs. asymmetric co-relevance estimation. Table 7 presents the comparison of the originally proposed asymmetric WBorda estimate with its symmetric version (see Equation 7). For the Regularization method, which relies on a symmetric co-relevance estimate, we present the results only for the symmetric version. We see that, with the exception of ClustMRF, the percentages for the asymmetric estimate are higher than those for its symmetric version. In addition, asymmetric WBorda is statistically significantly outperformed by the symmetric WBorda only in a few cases for ClustMRF. Furthermore, while both versions of WBorda outperform the reference comparisons in most cases, the asymmetric version does so (to a statistically significant degree) in a vast majority of the relevant comparisons. These findings attest to the merits of (i) integrating the measures used in WBorda to induce (either a symmetric or an asymmetric) co-relevance estimate; and (ii) using the originally proposed asymmetric WBorda in applications which call for asymmetric co-relevance estimation (namely, all those considered except Regularization).

Table 7: Comparing asymmetric with symmetric WBorda. The percentage of the 16 cases (8 experimental settings × 2 evaluation metrics) in which asymmetric (symmetric) WBorda outperforms (statistically significantly outperforms) symmetric (asymmetric) WBorda. And, the percentage of the 240 cases (8 experimental settings × 2 evaluation metrics × 15 reference comparisons excluding UBorda) in which WBorda outperforms (statistically significantly outperforms) the reference comparisons. In the first column percentages might not sum to 100 due to ties and rounding.

		Asymmetric/Symmetric		Reference	
		WBorda		Comparisons	
		Better	Sig. Better	Better	Sig. Better
NN test	Asymmetric	100.0	50.0	94.6	83.8
	Symmetric	0.0	0.0	86.7	77.1
Interpf	Asymmetric	87.5	18.8	85.0	43.3
	Symmetric	12.5	0.0	62.1	18.3
ClustMRF	Asymmetric	43.8	6.3	84.2	25.0
	Symmetric	43.8	12.5	85.8	27.9
ClustRanker	Asymmetric	68.8	31.3	88.3	59.2
	Symmetric	25.0	0.0	83.3	44.2
RWI	Asymmetric	50.0	31.3	99.6	83.8
	Symmetric	37.5	0.0	91.3	67.5
Regularization	Symmetric	–	–	92.9	66.3

6. CONCLUSIONS

We presented a method of estimating asymmetric co-relevance: the relevance of a document given another document assumed to be relevant. The co-relevance estimate is based on a learning-to-rank method that integrates various types of similarities with the query and the assumed relevant document, as well as query-independent document-quality measures. We showed that using our proposed estimate in three applications yields much better performance than that of using previously proposed co-relevance estimates. Studying how asymmetric co-relevance estimates can be used for results diversification is an interesting future venue.

Acknowledgments We thank the reviewers for their comments and Kripabandhu Ghosh for initial discussions. This work has been supported by and carried out at the Technion-Microsoft Electronic Commerce Research Center. This work has also been supported in part by Microsoft Research through its Ph.D. Scholarship Program.

7. REFERENCES

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

[2] J. A. Aslam and M. Frost. An information-theoretic measure for document similarity. In *Proc. of SIGIR*, pages 449–450, 2003.

[3] J. A. Aslam and M. Montague. Models for metasearch. In *Proc. of SIGIR*, pages 276–284, 2001.

[4] M. Bendersky, W. B. Croft, and Y. Diao. Quality-biased ranking of web documents. In *Proc. of WSDM*, pages 95–104, 2011.

[5] M. Bendersky and O. Kurland. Utilizing passage-based language models for ad hoc document retrieval. *Information Retrieval*, 13(2):157–187, 2010.

[6] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval*, 14(5):441–465, 2011.

[7] F. Diaz. Regularizing query-based retrieval scores. *Information Retrieval*, 10(6):531–562, 2007.

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

[9] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 28(5):1379–1389, 2001.

[10] N. Fuhr, M. Lechtenfeld, B. Stein, and T. Gollub. The optimum clustering framework: implementing the cluster hypothesis. *Information Retrieval*, 15(2):93–115, 2012.

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

[12] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proc. of SIGIR*, pages 76–84, 1996.

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

[14] T. Joachims. Training linear SVMs in linear time. In *Proc. of KDD*, pages 217–226, 2006.

[15] E. Krikon, O. Kurland, and M. Bendersky. Utilizing inter-passage and inter-document similarities for re-ranking search results. *ACM Transactions on Information Systems*, 29(1), 2010.

[16] O. Kurland. The opposite of smoothing: A language model approach to ranking query-specific document clusters. In *Proc. of SIGIR*, pages 171–178, 2008.

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

[18] O. Kurland and L. Lee. Corpus structure, language models, and ad hoc information retrieval. In *Proc. of SIGIR*, pages 194–201, 2004.

[19] O. Kurland and L. Lee. Respect my authority! HITS without hyperlinks utilizing cluster-based language models. In *Proc. of SIGIR*, pages 83–90, 2006.

[20] O. Kurland and L. Lee. PageRank without hyperlinks: Structural reranking using links induced by language models. *ACM Transactions on information systems*, 28(4):18, 2010.

[21] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. In *Language Modeling and Information Retrieval*, pages 1–10. Kluwer Academic Publishers, 2003.

[22] 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.

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

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

[25] X. Liu and W. B. Croft. Passage retrieval based on language models. In *Proc. of CIKM*, pages 375–382, 2002.

[26] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *Proc. of SIGIR*, pages 186–193, 2004.

[27] X. Liu and W. B. Croft. Experiments on retrieval of optimal clusters. Technical Report IR-478, Center for Intelligent Information Retrieval, University of Massachusetts, 2006.

[28] X. Liu and W. B. Croft. Representing clusters for retrieval. In *Proc. of SIGIR*, pages 671–672, 2006.

[29] X. Liu and W. B. Croft. Evaluating text representations for retrieval of the best group of documents. In *Proc. of ECIR*, pages 454–462, 2008.

[30] S.-H. Na. Probabilistic co-relevance for query-sensitive similarity measurement in information retrieval. *Information Processing and Management*, 49(2):558–575, 2013.

[31] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proc. of WWW*, pages 83–92, 2006.

[32] S. Paliwal and V. Pudi. Investigating usage of text segmentation and inter-passage similarities to improve text document clustering. In *Proc. of MLDM*, pages 555–565, 2012.

[33] F. Raiber and O. Kurland. Ranking document clusters using markov random fields. In *Proc. of SIGIR*, pages 333–342, 2013.

[34] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proc. of TREC*, 1994.

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

[36] K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 1. *Information Processing and Management*, 36(6):779–808, 2000.

[37] A. Tombros and C. J. van Rijsbergen. Query-sensitive similarity measures for information retrieval. *Knowledge and Information Systems*, 6(5):617–642, 2004.

[38] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[39] C. C. Vogt and G. W. Cottrell. Fusion via a linear combination of scores. *Information Retrieval*, 1(3):151–173, 1999.

[40] E. M. Voorhees. The cluster hypothesis revisited. In *Proc. of SIGIR*, pages 188–196, 1985.

[41] X. Wan. A novel document similarity measure based on earth mover's distance. *Information Sciences*, 177(18):3718–3730, 2007.

[42] X. Wan. Beyond topical similarity: a structural similarity measure for retrieving highly similar documents. *Knowledge and Information Systems*, 15(1):55–73, 2008.

[43] X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. In *Proc. of SIGIR*, pages 178–185, 2006.

[44] J. S. Whissell and C. L. A. Clarke. Improving document clustering using Okapi BM25 feature weighting. *Information Retrieval*, 14(5):466–487, 2011.

[45] J. S. Whissell and C. L. A. Clarke. Effective measures for inter-document similarity. In *Proc. of CIKM*, pages 1361–1370, 2013.

[46] P. Willett. Query specific automatic document classification. *International Forum on Information and Documentation*, 10(2):28–32, 1985.

[47] L. Yang, D. Ji, G. Zhou, Y. Nie, and G. Xiao. Document re-ranking using cluster validation and label propagation. In *Proc. of CIKM*, pages 690–697, 2006.

[48] 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.

[49] B. Zhang, H. Li, Y. Liu, L. Ji, W. Xi, W. Fan, Z. Chen, and W.-Y. Ma. Improving web search results using affinity graph. In *Proc. of SIGIR*, pages 504–511, 2005.