Relevance Ranking based on Query-Aware Context Analysis

Ali Montazeralghaem, Razieh Rahimi, and James Allan

Center for Intelligent Information Retrieval, University of Massachusetts Amherst, Amherst, MA, USA
{montazer, rahimi, allan}@cs.umass.edu

Abstract. Word mismatch between queries and documents is a long-standing challenge in information retrieval. Recent advances in distributed word representations address the word mismatch problem by enabling semantic matching. However, most existing models rank documents based on semantic matching between query and document terms without an explicit understanding of the relationship of the match to relevance. To consider semantic matching between query and document, we propose an unsupervised semantic matching model by simulating a user who makes relevance decisions. The primary goal of the proposed model is to combine the exact and semantic matching between query and document terms, which has been shown to produce effective performance in information retrieval. As semantic matching between queries and entire documents is computationally expensive, we propose to use local contexts of query terms in documents for semantic matching. Matching with smaller query-related contexts of documents stems from the relevance judgment process recorded by human observers. The most relevant part of a document is then recognized and used to rank documents with respect to the query. Experimental results on several representative retrieval models and standard datasets show that our proposed semantic matching model significantly outperforms competitive baselines in all measures.

Keywords: Semantic Matching · Local context · Retrieval model .

1 Introduction

In basic retrieval models such as BM25 [30] and the language modeling framework [29], the relevance score of a document is estimated based on explicit matching of query and document terms. These retrieval models have been improved in several directions; in this study, we focus on two of them: (1) semantic matching, and (2) simulating human relevance decision making.

First, different choices of words between the authors of documents and users interested in those documents impose the long-standing challenge of term mismatch between query and documents. Basic retrieval models suffer from the term mismatch problem, since semantically related terms do not contribute to the relevance scores of documents.

Several techniques have been developed to address the term mismatch problem, including query expansion [24,37,31,10,39], latent models [8,14,4], and retrieval using distributed representations of words [13]. Query expansion techniques using global or local analysis of documents have shown improvements in the performance of retrieval models; however, these techniques suffer from query-independent analysis of documents in a large corpus or query drift [7], respectively. Latent models have been used for matching queries and documents represented in latent semantic space. Although semantic matching is required for information retrieval, exact matching, especially when query terms are new or rare, still provides strong evidence of relevance [23]. Thus, these latent models alone do not perform well for information retrieval [3]. Translation models were initially proposed to address the term mismatch problem in the language modeling framework for information retrieval. These models estimate the likelihood that a query can be generated as a translation of a given document. Ganguly et al. [13] used word embedding [22,28] to estimate document language models through a noisy channel to address the term mismatch problem. Although there is a large body of research on semantic matching of terms in queries and documents, many studies fail to capture important IR heuristics such as proximity and term dependencies [21].

The second direction considers how people actually make relevance decisions. Relevance in almost all retrieval models is measured by comparing query terms with terms in the entire text of a document. This fundamental choice of input to scoring functions is not compatible with how a person perceives a document as relevant or non-relevant to his/her information need. This mismatch can lead to non-optimal performance of retrieval systems [18,19]. Kong et al. [38] described that a person first tries to locate pieces of a document that are likely to be related to the query. For each piece, the person then makes relevance decision based on the local context of the piece. If the piece is found to be relevant to the query, the document is judged as relevant, otherwise other pieces are considered for evaluation. Surprisingly little attention has been given to relevance ranking based on simulating how a person makes relevance decisions. Wu et al. [36] proposed a retrieval model simulating human relevance decision making and using context of query terms, however their model is not based on semantic similarity between query and the context of query terms.

Simulating human relevance decision making, we propose a novel model for semantic matching in information retrieval. The document's relevance to a query is thus estimated based on local contexts in the document. Local contexts for determining relevance consist of query terms' window-based pieces of text. These local contexts reduce the amount of texts considered for estimation of relevance to a query, while no information related to the query will be missed. Having local contexts, we compare each piece of text with the query based on both exact and semantic term matching. The proposed semantic matching model relaxes the assumption of independence between query terms to some extent, in that semantic similarity of terms in the local context of a query term is weighted by similarity of the query term with other query terms.

Our model can thus produce document ranking effectively and efficiently. Finally, our proposed model for relevance ranking provides the basis for natural integration of semantic term matching and local document context analysis into any retrieval model.

2 Related Work

2.1 Semantic Matching

In this section, we review the existing models for semantic term matching in information retrieval.

Query expansion. Query expansion has a long history in information retrieval, where a query is expanded with terms relevant to query terms. Query expansion can be done using global or/and local analysis [37]. Global expansion methods use word associations obtained independent of queries, such as corpus-wide information [16] or external resources such as WordNet [33]. Local query expansion methods mainly analyze the top retrieved documents for a query. Pseudo-relevance feedback is a well-known model of automatic query expansion, and has shown improvements in the performance of retrieval. Several models for pseudo-relevance feedback has been developed [31,20,24,26,2,25].

In addition, some models for query expansion use word embeddings [1,15]. Kuzi et al. find terms that are similar to the entire query or its terms using word embeddings and use them to expand query in the relevance model [17]. Diaz et al. [10] propose to use locally-trained embeddings for query expansion, where documents sampled from the top retrieved documents for queries are used to train word embeddings.

Latent models. Latent models represent queries and documents in a latent space of reduced dimensionality using the term-document matrix, such as LSA [9], PLSA [14], and LDA [4]. However, these models do not perform well for information retrieval [3]. Wei and Croft [35] use LDA topics to estimate document language models in the language modeling framework, and shown improvements in the performance of retrieval. Therefore, we compare our model with this LDA-based language model as a representative of this group.

Embedding-based retrieval models. Vulić and Moens [34] use (bilingual) word embeddings for monolingual and bilingual information retrieval, where queries and documents are represented as an average of the embeddings of their terms. However, the proposed model did not improve the performance of retrieval, unless it is combined with a basic retrieval model. Zheng and Callan [41] proposed a supervised model for re-weighting query terms in traditional retrieval models, BM25 and language modeling framework, based on embeddings of query terms. Although this model has shown to improve the performance of traditional retrieval models, the retrieval is still based on only exact matching of query and document terms.

Ganguly et al. [13] proposed a generalized estimate of document language models using a noisy channel, which captures semantic term similarities computed using word embeddings. In this model, words that are semantically related

to any word of a document contribute to the new estimate of document language models, while we only consider words that are semantically related to terms in local contexts of documents. Zamani and Croft [39] show that their query expansion model outperforms the generalized language model [13], therefore we only report the result of embedding-based estimation of query language models.

2.2 Local Context Analysis

Wu et al. [36] proposed a model for information retrieval by simulating how human makes relevance decisions. They consider context of query terms in documents to propose a novel retrieval method. They used related terms in context of query terms in document as expansion terms.

Kong et al. [38] introduced three principles for combining relevance evidence from different pieces of a document to make the final relevance decision: (1) Disjunctive Relevance Decision (DRD) principle, (2) Aggregate Relevance (AR) principle, and (3) Conjunctive Relevance Decision (CRD) principle. Following TREC guideline for relevance judgment stating that a document is relevant if any piece of it is relevant to the query.

2.3 Log-Logistic Retrieval Model

We briefly introduce the information-based retrieval model based on the Log-logistic distribution, which is used in our proposed ranking model. Document scores in this model is computed as follow:

$$\text{RSV}(Q, D) = \sum_{w \in Q} \text{count}(w, Q) \log(\frac{\text{tf}(w, D) + \lambda_w}{\lambda_w}), \tag{1}$$

where $\text{tf}(w, D) = \text{count}(w, D) \times \log(1 + c\frac{\text{avdl}}{|D|})$, c is a free parameter, avdl is average document length in the collection and $\lambda_w = \frac{N_w}{N}$ where N_w is the number of documents containing w and N is the number of documents in the collection.

3 The Proposed Semantic Model

The relevance ranking problem is to score a document $D = \{d_1, d_2, \ldots, d_n\}$ with respect to a query $Q = \{q_1, q_2, \ldots, q_m\}$ based on a combination of exact and semantic matching between query and document terms.

According to the TREC relevance judgment process, a document is relevant to a query if any piece of the document is relevant to the query[1]. Taking this definition, we first need to determine locations in the document that are likely related to the query.

Determining query-related locations. Wu et al. [36] proposed the query-centric assumption, which states that relevant information only occurs in the contexts around query terms in documents. The validity of this assumption is confirmed with a user study [18,19]. Following this assumption, a local context of query term q_i inside document D is denoted by $C(q_i, D)$ and is determined

[1] TREC: Text Retrieval Conference (TREC) data - English relevance judgements (2000) `https://trec.nist.gov/data/reljudge_eng.html`.

by a window around one occurrence of q_i in D. A symmetric window of size h centred at the occurrence of query term q_i in the document, gives a local context of following document terms.

$$C(q_i, D) = [d_{j-h}, \ldots, d_j = q_i, \ldots, d_{j+h}]. \tag{2}$$

Thus, each local context of a query term has a length of $2h + 1$. For simplicity, we refer to the local context of a query term in a document as the document context.

The next step is to estimate the relevance score of each query-centric context $c(q_i, D)$ with respect to query Q. For this purpose, a scoring function based on exact and semantic matching is desired. In addition, the scoring function should satisfy the constraints defined on information retrieval models so that ranking based on these scores provides reasonable rankings [12]. The Log-logistic model [5] has shown to be effective for information retrieval [6]. Therefore, we derive our local context scoring function based on the Log-logistic model as follows:

$$S(Q, C(q_i, D)) = \sum_{q_j \in Q} \log(\frac{\text{sim}(q_j, C(q_i, D)) + \lambda_{q_j}}{\lambda_{q_j}}), \tag{3}$$

where $\lambda_{q_j} = N_{q_j}/N$ is computed based on N_{q_j} representing the number of documents in the collection containing q_j and the total number of documents in the collection, N. This scoring function is obtained by replacing the normalized frequency of a query term in a document used in the Log-logistic scoring function of Eq. (1) by semantic similarity of the query term with terms in the document context. The similarity of term q_j with respect to the local context $C(q_i, D)$ which is a short text, is then estimated based on

$$\text{sim}(q_j, C(q_i, D)) = \sum_{w \in C(q_i, D)} \text{sim}(q_j, w) \times \mathbb{1}_{(\text{sim}(q_j, w) > \theta)}, \tag{4}$$

where $\mathbb{1}_{(.)}$ represents the indicator function taking on a value of 1 if the similarity between two terms is above the threshold parameter θ, and 0 otherwise. This indicator function is added to filter out the impacts of words unrelated to the query. Using this estimation of similarity, exact occurrences of query terms as well as words semantically related to query terms contribute to the relevance scores of local contexts in Eq. (3). And when parameter θ is set to the similarity value of a term to itself, the scoring function reduces to exact term matching.

The underlying assumption in the scoring function of Eq. (3) as well as many well-established retrieval models is independence between query terms. However, the similarity between query-centric (q_i) and current query term (q_i) does not consider in this function. As an example, consider the query 303 of TREC Robust dataset, "Hubble telescope achievements". And assume that we want to score a local context of query term "Hubble" which is a space telescope in a document. Thus, term "Hubble" is in the center of this local context. Although terms "Hubble" and "telescope" should logically have a higher similarity degree than terms "Hubble" and "achievements" in any term association resource, we

believe occurrences of terms related to "achievements" in the local context makes it more likely to be relevant to the query than those of "telescope", because we already know that "Hubble" exists in the context and occurring "telescope" can not have much information of relevance compared to "achievements". Therefore, to account for this observation, we proposed the following function to score a document context with respect to a query.

$$S(Q, C(q_i, D)) = \sum_{q_j \in Q} \log(\frac{\text{sim}(q_j, C(q_i, D)) + \lambda_{q_j}}{\lambda_{q_j}}) \times \text{dis}(q_i, q_j), \tag{5}$$

where $\text{dis}(q_i, q_j)$ denotes the semantic difference between the query term q_i in the center of the local context and the current query term q_j. We add $\text{dis}(q_i, q_j)$ to this function because we want to promote occurring query terms that semantically are far from query-centric. We compute $\text{dis}(q_i, q_j) = a - \text{sim}(q_i, q_j)$, where a is a constant that its value is obtained as $a = 1 + \text{sim}(t, t)$.

Herein, we use word embeddings to compute term similarities. Therefore, the similarity of a query term with a document context in Eq. (4) is computed as follows:

$$\text{sim}(q_j, C(q_i, D)) = \sum_{w \in C(q_i, D)} \cos(\boldsymbol{q_j}, \boldsymbol{w}) \times \mathbb{1}_{(\cos(\boldsymbol{q_j}, \boldsymbol{w}) > \theta)}, \tag{6}$$

where term vectors denote their embeddings in a continues space, and cos(.) function computes the cosine similarity between two vectors. Accordingly, the dissimilarity between query terms is computed as:

$$\text{dis}(q_i, q_j) = 2 - \cos(\boldsymbol{q_i}, \boldsymbol{q_j}). \tag{7}$$

the cosine similarity gives a value in the range of $[-1, 1]$ meaning that in case of perfect similarity ($\cos(\boldsymbol{q_i}, \boldsymbol{q_j}) = 1$) the dissimilarity is minimum $\text{dis}(q_i, q_j) = 1$ (note that because we use Eq.7 in Eq.5, the minimum value of dissimilarity should not be 0) and when $\cos(\boldsymbol{q_i}, \boldsymbol{q_j}) = -1$ dissimilarity is maximum i.e., $\text{dis}(q_i, q_j) = 3$.

Document relevance score. Obtaining the relevance score of each local context of query terms in a document, we then need to score the document with respect to the query based on the scores of its local contexts. We start by estimating the relevance score of a document with respect to each query term. Let $\zeta(q_i, D)$ denote the set of all local contexts of query term q_i in document D,

$$\zeta(q_i, D) = \{C_1(q_i, D), C_2(q_i, D), \ldots, C_k(q_i, D)\}, \tag{8}$$

where k equals to frequency of term q_i in the document, $\text{TF}(q_i, D)$.

The relevance scores of local contexts in the set $\zeta(q_i, D)$ should be aggregated to estimate the relevance score of document D with respect to q_i, denoted by $S_L(q_i, D)$. Following the aggregation principles introduced by Kong et al. [38], we consider two different aggregation function. The first *disjunctive relevance*

decision principle indicates that a document is relevant if any one of its local contexts is relevant to the query. Accordingly, we estimate $S_L(q_i, D)$ using

$$S_L(q_i, D) = \max_{C \in \zeta(q_i, D)} S(Q, C(q_i, D)). \tag{9}$$

Therefore, a document is scored according to its most relevant part to the query.

The second *aggregate relevance* principle states that a document with more pieces relevant to the query should get a higher relevance score. Following this relevance principle, we compute the relevance score of a document as

$$S'_L(q_i, D) = \sum_{C \in \zeta(q_i, D)} S(Q, C(q_i, D)). \tag{10}$$

The max aggregation function conforms to the TREC definition of document relevance[2]; a document is relevant if any part of the document is relevant, regardless of how small that part is compared to the entire document. We also observed better retrieval performance using this aggregation. Therefore, we adopt the max aggregation function for our proposed model and show the effectiveness of using the other function in one experiment.

Normalizing local relevance scores. Relevance scores of documents with respect to query terms in Eq. (9) are not theoretically bonded above, because we do not consider any normalization over semantic similarity of query term q_j with respect to the query-centric context in Eq 4.

A transformation function $f(.)$ to normalize local relevance scores of documents needs to satisfy three constraints: 1) vanishes at 0, 2) upper bounded to 1, 3) $f'(x) > 0$ to make sure that as the value of x increases, the output of function also increases. The simple yet effective function $f(x) = \frac{x}{x+\sigma}$ used in multiple information retrieval models [27] satisfies the three mentioned constraints. Therefore, to normalize the local relevance scores, we use this function as follows:

$$S_N(q_i, D) = \frac{S_L(q_i, D)}{S_L(q_i, D) + \sigma}, \tag{11}$$

where $\sigma > 0$ is a free parameter in this function.

Final document scores. Having the relevance score of a document with respect to each query term, the final score of the document can be calculated. For this purpose, we use weighted sum of scores of each query term to consider the importance of each query term in ranking. Therefore, the final score of a document is estimated as follow:

$$score(Q, D) = \sum_{q_i \in Q} S_N(q_i, D) \times \mathcal{W}(q_i, D), \tag{12}$$

where $\mathcal{W}(qi, D)$ is the importance of query term q_i given document D.

Query term importance. Importance of each query term has two sides: 1) global importance which is mainly computed by the inverse document frequency

[2] `https://trec.nist.gov/data/reljudge_eng.html`.

(idf) of the term in the collection. 2) local importance which can be the weight of the query term in the document. Existing retrieval models provide document ranking based on these two factors. Therefore, we can use any basic retrieval function to weigh query terms. In the experiments section, we show the results of using BM25, language modeling framework, and Log-logistic models for weighting query terms.

4 Discussion

Computational time: To compute the semantic similarity between query terms and document, we use the query-centric contexts. To do that, we first extract the positions of query terms in the document by Indri. Then, we find the neighbors of the query terms based on their positions in the document. By finding the neighbors of the query terms, we compute the semantic similarity between query terms and the query-centric contexts (i.e., the neighbors of the query terms). In other words, we do not consider all terms in a document to compute the semantic similarity. In contrast, the generalized language model [13] take into account all terms in the document to find the semantic similarity between the query and document which makes their approach so expensive. Note that our approach still is more effective since is more compatible with the human judgment process. Our approach is also more efficient compared to the topic modeling language model (LDA) [35], since they also use all terms in a document to model term associations. Also, the running time for each of the Gibbs sampling in LDA increases linearly with the number of documents N and the number of topics K i.e., $O(NK)$ which makes their approach even more expensive compared to the generalized language model.

Zamani and Croft [39] proposed an embedding query expansion named EQE1 to estimate query language model. They used word embedding similarities to find terms in the entire vocabulary that are semantically related to the query terms. This method is much faster than the previous ones (i.e., LDA model [35] and generalized language model [13]) but is not optimal since it needs to compute similarity scores between query terms and all terms in the vocabulary.

Properties of the proposed method: Fang et al. [11], proposed seven constraints for IR models and showed that it is necessary to satisfy them to get good performance. To compute the semantic similarity score between a query-centric context and query terms, it is also necessary to satisfy these constraints. For example, if we have a query with two terms, and two query-centric contexts, the context that can interpret more distinct query words should be assigned a higher score. Clinchant et al. [6] showed that Log-Logistic model satisfies all constraints in the PRF framework. Therefore, we modify this model in our approach to compute semantic similarity between query-centric context and query terms. In other words, by modifying Log-Logistic model and considering query-centric context as a small document, we make sure that our model satisfies all constraints proposed by Fang et al. [11].

Tao and zhai [32], proposed proximity based constraints for IR models. There are similar constraints for pseudo relevance feedback [25]. They showed that by comparing two documents that match the same number of query words, it is

more desirable to rank the document in which all query terms are close to each other above another one. We argue that this assumption also is valid in semantic space. By using the query-centric window, we implicitly promote documents that have query terms close in semantic or exact matching. Therefore, our method can capture important IR characteristics i.e. exact/semantic matching signals, proximity heuristics, and query term importance.

5 Experiments

In this section, we aim to address the following research questions.

RQ1: How does our model perform with different retrieval functions to weight query terms? (see Eq.12)

RQ2: How does our model perform compared to existing retrieval models?

RQ3: What is the effect of different principle functions for combining relevance evidence, including Aggregate Relevance (AR) principle and Disjunctive Relevance Decision (DRD), in our model?

Experimental setup. We use three standard TREC collections in our experiments: Robust, Gov2, and WT10g. We use the title of topics as queries. We use standard INQUERY stopword list to remove stopwords and no stemming is performed. The experiments are carried out using the Indri and the Lemur toolkits[3]. In our experiments, we use pre-trained word embeddings of Glove, trained on Wikipedia 2014 and Gigaword 5 corpus [28]. Statistically significant differences of performances are determined using the two-tailed paired t-test computed at a 95% confidence level over average precision per query.

Baseline methods. We compare our proposed model with three categories of existing models for information retrieval: basic retrieval models, retrieval models that consider semantic similarity or topic modeling, and proximity-based retrieval models. Since our proposed model is an unsupervised semantic matching, we do not compare our model with supervised approaches and neural network based models that require labeled data. Three basic retrieval models are used as baseline: **BM25:** An effective and widely-used retrieval method [30]. **LM:** Standard language modeling approach for information retrieval where document language models are smoothed using the Jelinek-Mercer smoothing method [40]. **Logistic:** An information-based retrieval method using the Log-logistic distribution [6]. For proximity-based retrieval models, we choose the sequential dependence model (**SDM**) as a baseline, which considers term dependencies in the language modeling framework using Markov random fields [21]. Baseline models that take advantages of semantic similarity or topic modeling are: **EQE1:** An embedding-based model for query expansion proposed by Zamani and Croft [39]. EQE1 is shown to outperform other embedding-based expansion models, e.g., embedding-based expansion of document language models [13] and heuristic-based query expansion using word embeddings [1]. Therefore, we only compare our model with the EQE1 model in the experiments. Note that in this experiment, we only consider methods that select expansion terms based on word

[3] `http://lemurproject.org/`

Table 1: Performance of the proposed method with different retrieval models as the weighing function in Eq. (12). ▲ indicates that the improvements over corresponding retrieval model are statistically significant.

Dataset	Metric	LM	LCD-LM	BM25	LCD-BM25	Logistic	LCD-Logistic
Robust	*MAP*	0.2048	0.2351▲	0.2264	0.2458▲	0.2258	0.2481▲
	P@10	0.3566	0.3916▲	0.4088	0.4249▲	0.4004	0.4329▲
	nDCG@10	0.3580	0.3948▲	0.4171	0.4327▲	0.4132	0.4391▲
	Recall	0.5093	0.5501▲	0.5156	0.5507▲	0.5195	0.5539▲
	GMAP	0.1043	0.1324▲	0.1235	0.1432▲	0.1257	0.1462▲
WT10g	*MAP*	0.1119	0.1469▲	0.1773	0.1899▲	0.1744	0.1945▲
	P@10	0.1806	0.2224▲	0.2622	0.2867▲	0.2704	0.2908▲
	nDCG@10	0.1695	0.2239▲	0.2904	0.3132▲	0.2842	0.3110▲
	Recall	0.5430	0.5896▲	0.5883	0.5989	0.5792	0.6058▲
	GMAP	0.0400	0.0646▲	0.0680	0.0847▲	0.0656	0.0871▲
Gov2	*MAP*	0.2521	0.2850▲	0.2671	0.2937▲	0.2679	0.2926▲
	P@10	0.5332	0.5672▲	0.5669	0.5858▲	0.5479	0.5655▲
	nDCG@10	0.4343	0.4578	0.4681	0.4907▲	0.4535	0.4739▲
	Recall	0.6028	0.6336▲	0.6188	0.6487▲	0.6166	0.6534▲
	GMAP	0.1701	0.1946▲	0.1897	0.2096▲	0.1863	0.2126▲

embeddings and not other information sources such as the top retrieved documents for each query (PRF). **LDA:** An LDA-based estimation of document language models [35].

We compare baseline models with some variants of our proposed model. Variants of our model starting with prefix **LCA** or **LCD** indicate that documents are scored using aggregate relevance or disjunctive relevance, respectively.

Parameter setting. In all experiments, parameter c in Eq. 1, σ in Eq. 11, θ in Eq. 6, the parameter λ of the smoothing method of the LM baseline, and parameters b and k_1 of the BM25 baseline are set using 2-fold cross validation to optimize MAP performance over the queries of each collection. The value of parameters c, σ, and θ are selected from $\{1, 3, 6, \cdots, 12\}$, $\{1, 5, 10, \cdots, 20\}$, and $\{0.3, 0.4, 0.5, \cdots, 1.0\}$, respectively. The parameter λ of the LM baseline is swept between $\{0, 0.1, \cdots, 1\}$ and the value of b and k_1 of BM25 baseline are chosen from $\{0.75, 1.0\}$ and $\{1.0, 2.0\}$, respectively. In all experiments, the dimensions of embedding vectors is 200. We set the LDA hyper-parameters α and β to $50/K$ and 0.001, respectively, where K is the number of topics in LDA. K is set to 800 as suggested in [35]. For the SDM model, the weight of the unigram component, the ordered and unordered window are selected from $\{0, 0.1, \cdots, 1\}$. We also made sure that they sum to 1.

5.1 Effectiveness of different weighting functions

In this section, without loss of generality, we use three simple but effective retrieval functions including LM, BM25, and Log-Logistic in our model to weight query terms in Eq. (12), which aims to answer **RQ1**. The results of this experiment are reported in Table 1. According to this table, semantic matching in the local contexts of documents improves retrieval effectiveness in all cases. The

Table 2: Performance of proposed method and baselines. The superscript ▲ indicates that the improvements over all other baselines are statistically significant.

Dataset	Metric	LM	BM25	Logistic	SDM	LDA	EQE1	LCD-Logistic
Robust	*MAP*	0.2048	0.2264	0.2258	0.2273	0.2299	0.2278	**0.2481** ▲
	P@10	0.3566	0.4088	0.4004	0.3936	0.4123	0.4040	**0.4329** ▲
	nDCG@10	0.3580	0.4171	0.4132	0.4069	0.4192	0.4094	**0.4391** ▲
	Recall	0.5093	0.5156	0.5192	0.5188	0.5209	0.5244	**0.5539** ▲
	GMAP	0.1043	0.1235	0.1257	0.1212	0.1187	0.1186	**0.1462** ▲
WT10g	*MAP*	0.1119	0.1773	0.1744	0.1845	-	0.1867	**0.1945** ▲
	P@10	0.1806	0.2622	0.2704	0.2776	-	0.2750	**0.2908** ▲
	nDCG@10	0.1695	0.2904	0.2842	0.2916	-	**0.3110**	**0.3110**
	Recall	0.5430	0.5883	0.5792	0.5930	-	0.6046	**0.6058**
	GMAP	0.0400	0.0680	0.0656	0.0783	-	0.0792	**0.0871** ▲
Gov2	*MAP*	0.2521	0.2671	0.2679	0.2695	-	0.2731	**0.2926** ▲
	P@10	0.5332	0.5669	0.5479	0.5608	-	**0.5682**	0.5655
	nDCG@10	0.4343	0.4681	0.4535	0.4615	-	0.4671	**0.4739** ▲
	Recall	0.6028	0.6188	0.6166	0.6250	-	0.6232	**0.6534** ▲
	GMAP	0.1701	0.1897	0.1863	0.1859	-	0.1901	**0.2126** ▲

improvements are statistically significant in most cases. This shows the effectiveness of our approach in integration of semantic matching into retrieval models. One can also observe that using Log-Logistic function for weighting the importance of query terms in our model outperforms using BM25 or the language model framework in most cases. This observation demonstrates that relevance scores of local contexts are more compatible with the scores of the Log-logistic model. Therefore, in the next experiments, we use this retrieval model as the weighing function in our model.

Table 3: Comparing different principle function for combining local context relevance scores on Gov2 only.

Metric	LCA-Logistic	LCD-Logistic
MAP	0.2730	**0.2926**
P@10	0.5628	**0.5655**
nDCG@10	0.4585	**0.4739**
Recall	0.6297	**0.6534**
GMAP	0.1897	**0.2126**

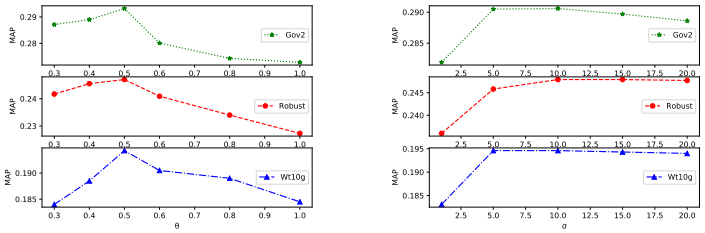
5.2 Performance of the Proposed Model

In this section, we compare our model with the baselines, which aims to address **RQ2**. The results of this experiment are reported in Table 2. According to this table, the proposed method (i.e., LCD-Logistic) outperforms SDM. This shows that our model in addition to using the proximity of query terms in a document improves the retrieval performance by exploiting the semantic similarity of terms. Unlike SDM, the EQE1 and LDA baseline models consider semantic similarity and topic modeling in document ranking, respectively. According to the results in this table, LCD-Logistic outperforms EQE1 and LDA-based retrieval models. We only report the results of the LDA model on Robust due to the prohibit

training time of LDA on the other two collections. These comparisons show the importance of capturing semantic similarity in the local context for information retrieval. We also report GMAP to evaluate our method in confrontation with hard queries. We also report the recall metric in this table. According to the results, LCD-Logistic improves recall in Robust and Gov2 collections substantially.

5.3 Different principle functions for combining relevance evidences

This section aims to answer **RQ3**. The results of this experiment are shown in Table 3[4]. Comparing LCD and LCA variants of our model together shows that the LCD variant outperforms LCA in all cases. This means that using the most relevant local context of a document to score it, following the disjunctive relevance principle, has better performance.



(a) (b)

Fig. 1: Sensitivity of the proposed method (LCD-Logistic) to the θ and σ

Parameter Sensitivity Analysis: Figure 1a shows the sensitivity of the proposed method to the similarity threshold in Eq 4. According to this figure, the best value for the similarity threshold is 0.5 in all datasets. It is worth noting that by setting the value of θ to 1, we just consider exact matching in the local context of query terms. Figure 1b shows sensitivity of LCD-Logistic to the normalization parameter σ in Eq 11. According to this figure, the best value for this parameter in all collections is 10.

6 Conclusion

We propose a new model for semantic matching in information retrieval. Our model is designed based on simulating human judgment process to find high-quality similarity scores between query and document. The proposed method is designed to be able to capture important IR heuristics, e.g, proximity of query terms in documents, semantic matching between query and document terms, and importance of query terms. We showed that our model can be integrated into any retrieval models and improve their performance significantly.

Acknowledgements. This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-1617408. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

[4] Note that for the sake of space, in this experiment, we just consider the Gov2 collection.

References

1. ALMasri, M., Berrut, C., Chevallet, J.P.: A comparison of deep learning based query expansion with pseudo-relevance feedback and mutual information. In: Advances in Information Retrieval (2016)
2. Ariannezhad, M., Montazeralghaem, A., Zamani, H., Shakery, A.: Iterative estimation of document relevance score for pseudo-relevance feedback. In: European Conference on Information Retrieval. pp. 676–683. Springer (2017)
3. Atreya, A., Elkan, C.: Latent semantic indexing (lsi) fails for trec collections. SIGKDD Explor. Newsl. 12(2), 5–10 (Mar 2011)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (Mar 2003)
5. Clinchant, S., Gaussier, E.: Information-based models for ad hoc ir. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. pp. 234–241. ACM (2010)
6. Clinchant, S., Gaussier, E.: A theoretical analysis of pseudo-relevance feedback models. pp. 6:6–6:13. ICTIR '13 (2013)
7. Collins-Thompson, K.: Reducing the risk of query expansion via robust constrained optimization. In: Proceedings of the 18th ACM conference on Information and knowledge management. pp. 837–846. ACM (2009)
8. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American society for information science 41(6), 391–407 (1990)
9. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. Technol. 41, 391–407 (1990)
10. Diaz, F., Mitra, B., Craswell, N.: Query expansion with locally-trained word embeddings. arXiv preprint arXiv:1605.07891 (2016)
11. Fang, H., Tao, T., Zhai, C.: A formal study of information retrieval heuristics. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 49–56. ACM (2004)
12. Fang, H., Tao, T., Zhai, C.: Diagnostic evaluation of information retrieval models. ACM Trans. Inf. Syst. 29(2), 7:1–7:42 (Apr 2011)
13. Ganguly, D., Roy, D., Mitra, M., Jones, G.J.: Word embedding based generalized language model for information retrieval. pp. 795–798. SIGIR '15 (2015)
14. Hofmann, T.: Probabilistic latent semantic indexing. pp. 50–57. SIGIR '99 (1999)
15. Imani, A., Vakili, A., Montazer, A., Shakery, A.: Deep neural networks for query expansion using word embeddings. In: European Conference on Information Retrieval. pp. 203–210. Springer (2019)
16. Jones, K.S.: Automatic keyword classification for information retrieval. The Library Quarterly 41(4), 338–340 (1971)
17. Kuzi, S., Shtok, A., Kurland, O.: Query expansion using word embeddings. In: Proceedings of the 25th ACM international on conference on information and knowledge management. pp. 1929–1932. ACM (2016)
18. Li, X., Liu, Y., Mao, J., He, Z., Zhang, M., Ma, S.: Understanding reading attention distribution during relevance judgement. pp. 733–742. CIKM '18 (2018)
19. Li, X., Mao, J., Wang, C., Liu, Y., Zhang, M., Ma, S.: Teach machine how to read: Reading behavior inspired relevance estimation. SIGIR (2019)
20. Lv, Y., Zhai, C.: A comparative study of methods for estimating query language models with pseudo feedback. In: Proceedings of the 18th ACM conference on Information and knowledge management. pp. 1895–1898. ACM (2009)

21. Metzler, D., Croft, W.B.: A markov random field model for term dependencies. pp. 472–479. SIGIR '05 (2005)
22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
23. Mitra, B., Diaz, F., Craswell, N.: Learning to match using local and distributed representations of text for web search. In: Proceedings of the 26th International Conference on World Wide Web. pp. 1291–1299. International World Wide Web Conferences Steering Committee (2017)
24. Montazeralghaem, A., Zamani, H., Shakery, A.: Axiomatic analysis for improving the log-logistic feedback model. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 765–768 (2016)
25. Montazeralghaem, A., Zamani, H., Shakery, A.: Term proximity constraints for pseudo-relevance feedback. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1085–1088 (2017)
26. Montazeralghaem, A., Zamani, H., Shakery, A.: Theoretical analysis of interdependent constraints in pseudo-relevance feedback. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 1249–1252 (2018)
27. Paik, J.H.: A novel tf-idf weighting scheme for effective ranking. pp. 343–352. SIGIR '13 (2013)
28. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (Oct 2014)
29. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. pp. 275–281 (1998)
30. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: SIGIR'94. pp. 232–241. Springer (1994)
31. Salton, G.: The SMART Retrieval System—Experiments in Automatic Document Processing. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1971)
32. Tao, T., Zhai, C.: An exploration of proximity measures in information retrieval. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 295–302. ACM (2007)
33. Voorhees, E.M.: Query expansion using lexical-semantic relations. pp. 61–69. SIGIR '94 (1994)
34. Vulić, I., Moens, M.F.: Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. pp. 363–372. SIGIR '15 (2015)
35. Wei, X., Croft, W.B.: Lda-based document models for ad-hoc retrieval. pp. 178–185. SIGIR '06 (2006)
36. Wu, H.C., Luk, R.W., Wong, K.F., Kwok, K.: A retrospective study of a hybrid document-context based retrieval model. Information processing & management 43(5), 1308–1331 (2007)
37. Xu, J., Croft, W.B.: Query expansion using local and global document analysis. pp. 4–11. SIGIR '96 (1996)
38. Y. Kong, K., Luk, R., Lam, W., S. Ho, K., Chung, F.L.: Passage-based retrieval based on parameterized fuzzy operators. In: The SIGIR 2004 Workshop on Mathematical/Formal Methods for Information Retrieval (2004)

39. Zamani, H., Croft, W.B.: Embedding-based query language models. pp. 147–156. ICTIR '16 (2016)
40. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: ACM SIGIR Forum. vol. 51, pp. 268–276. ACM (2017)
41. Zheng, G., Callan, J.: Learning to reweight terms with distributed representations. pp. 575–584. SIGIR '15 (2015)