Noname manuscript No.
(will be inserted by the editor)

Extending the Language Modeling Framework for Sentence Retrieval to include Local Context

Ronald T. Fernández · David E. Losada ·
Leif A. Azzopardi

the date of receipt and acceptance should be inserted later

Abstract Employing effective methods of sentence retrieval is essential for many tasks in Information Retrieval, such as summarization, novelty detection and question answering. The best performing sentence retrieval techniques attempt to perform matching directly between the sentences and the query. However, in this paper, we posit that the local context of a sentence can provide crucial additional evidence to further improve sentence retrieval. Using a Language Modeling Framework, we propose a novel reformulation of the sentence retrieval problem that extends previous approaches so that the local context is seamlessly incorporated within the retrieval models. In a series of comprehensive experiments, we show that localized smoothing and the prior importance of a sentence can improve retrieval effectiveness. The proposed models significantly and substantially outperform the state of the art and other competitive sentence retrieval baselines on recall-oriented measures, while remaining competitive on precision-oriented measures. This research demonstrates that local context plays an important role in estimating the relevance of a sentence, and that existing sentence retrieval language models can be extended to utilize this evidence effectively.

Keywords Information Retrieval, Sentence Retrieval, Language Models, Context

Ronald T. Fernández
Grupo de Sistemas Inteligentes
Universidad de Santiago de Compostela, Spain
Tel.: +34 981-563100 x13569
Fax: +34 981-528012
E-mail: ronald.teijeira@usc.es

David E. Losada
Grupo de Sistemas Inteligentes
Universidad de Santiago de Compostela, Spain
Tel.: +34 981-563100 x13570
Fax: +34 981-528012
E-mail: david.losada@usc.es

Leif A. Azzopardi
Department of Computing Science
University of Glasgow, UK
Tel: +44 (0)141-330 1631
Fax: +44 (0)141-330 4913
E-mail: leif@dcs.gla.ac.uk

1 Introduction

The sentence retrieval (SR) task consists of finding relevant sentences from a document base given a query. This task is very useful in a wide range of Information Retrieval (IR) applications, such as summarization, question answering, and opinion mining. SR is a challenging problem area that has attracted a great deal of attention recently [1, 29, 18, 16, 13]. The bulk of SR methods proposed in the literature are a straight-forward adaptation of standard retrieval models (such tf-idf, BM25, Language Models, etc), where the sentence is the unit of retrieval, as opposed to the document. This leads to SR models which estimate relevance based only on the match between query and sentence terms. The state of the art SR method is known as term frequency - inverse sentence frequency (tfisf) which is analogous to the traditional tf-idf method used in document retrieval [1, 13]. While, numerous attempts to develop more sophisticated models that employ techniques, such as Natural Language Processing and Clustering have been proposed [11, 8, 33], they have failed to significantly and consistently outperform the tfisf method. Consequently, little progress has been made in terms of improving sentence retrieval effectiveness.

To develop a more effective sentence retrieval method, we argue that the assumption engaged as a result of the naive application of document retrieval, i.e. that all sentences are independent, does not hold. This is because a sentence is surrounded by other sentences which help to contextualize it. Also the sentence is part of a document, and this sentence may or may not be important in representing the topic of the document. Presently, this *local context* is either ignored or underutilized by existing methods. We posit that by incorporating the local context within SR models, more effective SR methods can be developed.

The reasons for this are as follows: Any model using only standard term statistics to match query and sentences will suffer severely from the vocabulary mismatch problem because there is little overlap between the query and sentence terms. Intuitively, the local context could be used to improve retrieval, by helping to mitigate the difficulties posed by the vocabulary mismatch rooted in the sparsity of sentences. Additionally, current methods do not exploit the importance of a sentence in a document, which we posit is an important factor in determining the relevance of a sentence. A relevant sentence needs to be indicative of the query topic, but also representative and important in the context of the document, i.e. we assume that key statements within a document are more likely to be relevant.

To this aim, we propose a novel reformulation of the SR problem that includes the local context in a Language Modeling (LM) framework. Within this principled framework, it is possible to naturally include additional evidence into the smoothing process in order to enrich the representation of sentences. Also, the model provides a way to include a query-independent probability that encodes the importance of a sentence in a document. In a set of experiments performed over several TREC test collections, we compare the proposed models against existing SR models and demonstrate that using local context within a LM framework delivers retrieval performance that significantly outperforms the current state of the art in sentence retrieval.

The remainder of this paper is organized as follows. Section 2 presents previous work related to this research. Section 3 explains the methods we propose to address the SR problem. Section 4 reports on the conducted experiments and analyzes the outcomes. The paper concludes with Section 5, where a summary of our findings and directions for future work are presented.

2 Related Work

In this paper, we adopt the same definition of the sentence retrieval problem as proposed in the TREC Novelty Tracks [5,28,27]. Although these tracks are mostly focused on researching redundancy filtering, they also involve a SR task that enables research into how to retrieve sentences that are relevant to a given query.

As previously mentioned, there have been numerous SR methods that have been proposed in the literature. One of the first methods was coined as tfisf [1]. It is an adaptation of the document retrieval method tf-idf, but at the sentence level. This simple approach is regarded as the state of the art in SR as it has been shown to consistently outperform other methods [1,16,4]. As a matter of fact, this parameter-free method has been shown to perform at least as well as the best performing empirically tuned and trained SR models based on BM25 or LMs [16,4]. While this tends not to be the case in document retrieval, on other tasks where the unit of retrieval is smaller such as passage retrieval, vector-space models have performed empirically well. For instance, Kaszkiel and Zobel [9,10] showed that some cosine and pivoted models are highly effective for document ranking based on passages. Although we evaluate here SR (rather than document retrieval), past studies on passage-based document retrieval confirm also that vector-space methods are also state of the art models for query-passage scoring.

In [11], Li and Croft analyzed the components of sentences and identified patterns (such as phrases, name entities and combination of query terms) to estimate the relevance of the sentences. Although this method succeeded in detecting redundant information, it was not able to improve the tfisf baseline to estimate relevance. Clustering methods have been also considered as alternative techniques to improve SR models, such methods have shown mixed performance [8,33] seldom improving upon the tfisf baseline. These cluster methods also incur additional computation costs and increased complexity making them unattractive to implement. Query expansion techniques have been also proposed to improve the performance of current sentence retrieval approaches. Among them, the most common is query expansion via pseudo-relevance feedback [3,13] and with selective feedback [7,16], or relevance models [12]. While query expansion techniques tend to improve performance by addressing the vocabulary mismatch problem, they rely on good performance during the first pass of retrieval to realize such improvements.

In this paper, we reformulate the problem of sentence retrieval within the LM framework, where localized smoothing is employed to improve the representation of sentences. The work most related to this research has been performed by Losada and Fernández [16] and Murdock [18]. In [16], the local context of a sentence was informally introduced into the computation of sentence similarity. Basically, extra weight was given to those terms that have high frequency in the associated documents. In [18], the estimation of the sentence language model included some local context, and combines the evidence from the sentence and document level. More specifically, a simple mixture model of the sentence, document and collection was proposed in order to form a better representation of the sentence. From the limited experiments reported, Murdock showed that the mixture model was better than other LM methods with the TREC novelty data. However, the results are far from conclusive because competitive SR methods, such as tfisf, were not evaluated. Nor was any indication of the sensitivity of the method w.r.t the smoothing parameters reported. In this paper, we provide a more general framework that encompasses both previous formulations using Language Models, but also provides avenues for incorporating other forms of local context.

3 Sentence Retrieval Models

The SR task consists of estimating the relevance of each sentence s in a given document set, and supplying the user with a ranked list of sentences that satisfy his/her need (expressed as a user query q). In this section, we first outline the standard LM approach applied to the problem of SR. Then, we propose a novel reformulation which includes local context seamlessly and intuitively within the model. Finally, we conclude the section with a description of baseline SR models (tfisf and BM25).

3.1 Sentence Retrieval with Language Models (Standard Method)

Language Models are probabilistic mechanisms to explain the generation of text [19]. The simplest LM is the unigram LM, which consists of associating a probability to each word of the vocabulary [31, 6, 17]. This is a very intuitive and powerful approach that has been shown to be very effective in many IR tasks, such as ad-hoc retrieval [31], distributed IR [24], and expert finding [2].

Given the SR problem, the idea is to estimate relevance according to the probability of generating a sentence s given the query q, expressed as $p(s|q)$. Instead of directly estimating this probability, Bayes Theorem is applied, and sentences can be ranked using the query-likelihood approach, $p(q|s)$[1]. The probability of a query q given the sentence s can then be estimated using the standard LM approach, where for each sentence s, a sentence LM is inferred. From the sentence model θ_s it is assumed that each query term t is sampled independently and identically, such that:

$$p(q|\theta_s) = \prod_{t \in q} p(t|\theta_s)^{c(t,q)} \tag{1}$$

where, $c(t,q)$ is the number of times the term t appears in q. The sentence model is constructed through a mixture between the probability of a term in the sentence and the probability of a term occurring in some background collection (i.e. maximum likelihood estimators of sentence and collection, respectively). This is usually performed in one of two ways by using (a) Jelinek-Mercer (JM) smoothing as shown in Equation 2, or (b) Dirichlet (DIR) smoothing as shown in Equation 3.

$$p(t|\theta_s) = (1-\lambda)p(t|s) + \lambda p(t) \tag{2}$$

$$p(t|\theta_s) = \frac{c(t,s) + \mu p(t)}{c(s) + \mu} \tag{3}$$

where $c(t,s)$ is the number times that t appears in s, and $c(s)$ is the number of terms in the sentence. λ and μ are parameters that control the amount of smoothing. Note that, in Equations 2 and 3, the smoothing expression ignores any local context and resorts immediately to the most general background knowledge $p(t)$. This is a strong assumption because it focuses the computation on sentence and collection statistics, without regard to any reference to other terms and phrases in sentences within the same document. As previously mentioned, many SR models [1] take similar simplifications as the query-sentence similarity values do not take into account any information from the document (i.e. all sentences are treated independently).

[1] This assumes that there is not a priori preference for particular types of sentences, i.e. $p(s)$ is uniform.

JM and DIR smoothing yield to retrieval matching functions with specific length retrieval trends. In [14] and [26], the authors studied these trends. In [14], Losada and Azzopardi reported that DIR smoothing performs better than JM smoothing by showing that the document length pattern resembles the relevance pattern. They showed that DIR priors balance the query modeling and the document modeling roles, whereas JM smoothing does not consider the document length in the smoothing process. Thus, JM leads to poor retrieval performance because documents tend to be longer than the documents retrieved by DIR and the smoothing cannot compensate this. In [26], Smucker and Allan demostrated that DIR smoothings performance advantage arises from an implicit document prior that favors longer documents by smoothing them less. They tested the performance of a DIR prior and the JM smoothing with and without the document prior and showed that both methods smooth documents identically, except that the DIR prior smooths longer documents less. The result of this meant that the DIR prior tends to favor the retrieval of longer documents. Given the sentence retrieval problem, it is an open question as to what kind of length correction is appropriate for this task and whether the implicit length correction of smoothing methods employed help or hinder in the retrieval of relevant sentences.

3.2 Sentence Retrieval using Language Models with Local Context

In this section, we relax the independence assumption between sentences and assume that the document (i.e. the local context) plays an important role in determining the relevance of a sentence. Therefore, we treat the SR problem as a problem of estimating the probability of the query and the document given the sentence, i.e. is the sentence likely to be a generator of both the query and the document? This assumes that there is a correlation between this likelihood, $p(q,d|s)$ (where d is the document that contains s) and the relevance of the sentence. Thus, we posit that relevance is affected by how well the sentence explains both the document and the query topic (as opposed to the query topic alone). In order to simplify the estimation of the conditional joint probability, we can rewrite it as follows:

$$p(q,d|s) = p(q|s,d)p(d|s) \tag{4}$$

where $p(q|s,d)$ is the probability of the query given the sentence and document, and $p(d|s)$ is the probability of the document given the sentence. Now we can clearly see that the estimation of the query likelihood will depend on both the sentence and the document. In addition, the $p(d|s)$ provides another way in which the local context is captured, by encoding the importance of a sentence within the document. In the next subsections we consider how these probabilities can be estimated.

3.3 Estimating $p(d|s)$

The probability of generating the document given the sentence, $p(d|s)$, can be regarded as a measure of the importance of the sentence within the topic of the document. Formally, this expression can be rewritten using Bayes' rule:

$$p(d|s) = \frac{p(s|d)p(d)}{p(s)} \tag{5}$$

where $p(s|d)$ is the probability of a sentence given a document, the $p(s)$ the probability of a sentence, and $p(d)$ is the prior probability of a document. Here, we assume that there is no a priori preference towards any of the documents, and treat $p(d)$ as a constant[2]. The $p(s|d)$ represents how likely the sentence is to be generated from the document, whereas $p(s)$ represents how likely the sentence is to be generated randomly. The ratio between the two expresses the importance of the sentence. Hence, in order to estimate $p(d|s)$, we compute $p(s)$ as:

$$p(s) = \prod_{t \in s} p(t)^{c(t,s)} \tag{6}$$

where $p(t)$ can be calculated using the maximum likelihood estimator of the term in a large collection: $p(t|C)$ (where C is the collection). Analogously, we define the probability of a sentence s given a document d as:

$$p(s|d) = \prod_{t \in s} p(t|d)^{c(t,s)} \tag{7}$$

where $p(t|d)$ is the probability of generating t from the maximum likelihood estimator of the document, and $c(t,s)$ usually equals one as most terms only appear once in a sentence (unless the term is a stop word). It is to be noted that the problem of obtaining null probabilities from these estimates does not exist because terms that occur in a sentence will have non-zero probability in the LM of the document. Observe that $p(d|s)$ will give preference to those sentences that are central to the document's topics (i.e. high $p(s|d)$) but also rare within the collection (i.e. low $p(s)$). In this paper we carefully study the effect of $p(d|s)$ on performance, and have designed a complete set of experiments where we compare the estimation described above against the simplest (and naive) assumption: $p(d|s)$ is uniform.

3.4 Estimating $p(q|s,d)$

In order to estimate the query likelihood given the sentence and the document, we do this in a similar manner to the standard approach: first we assume that there is a model $\theta_{s,d}$ which generates the query terms, such that the probability of query given the sentence and the document is:

$$p(q|s,d) = \prod_{t \in q} p(t|\theta_{s,d})^{c(t,q)} \tag{8}$$

The LM $p(t|\theta_{s,d})$ is determined by the sentence and the local context denoted by d, thus we can represent the model as a mixture between the probability of a term in the sentence and the probability of a term in a document, which is then smoothed by the background model. The idea is that the terms in the document provide meaning to the sentence, and can improve the estimate of the relevance of a sentence.

For the time being, we assume that $p(t|d)$ is the normalized term frequency of t in d, but later we explore restricting this estimate to the sentences surrounding the sentence s.

There are several ways in which a mixture model can be defined using smoothing:

[2] A simple alternative, which could be explored as part of future work, would be to estimate the prior based on the estimated relevance of the document.

Three Mixture Model (3MM): The first model we propose here is a mixture of three LMs. This model assumes that queries are generated from a mixture of three different probability distributions: a LM for the sentence, $p(t|s)$, a LM for the document, $p(t|d)$, and a LM for the collection, $p(t|C)$ (or, simply, $p(t)$). Formally, we define this approach as:

$$p(t|\theta_{s,d}) = \lambda p(t|s) + \gamma p(t|d) + (1 - \lambda - \gamma)p(t) \tag{9}$$

where λ and γ are smoothing parameters such that $\lambda, \gamma \in [0,1]$. This estimator was initially proposed by Murdock in [18]. Other authors have also applied 3MMs for other tasks such as question-answering [30]. Since the 3MM is very general, it is worth considering alternatives which smooth the sentence with the document and the collection but in a length-dependent way. This can be achieved by either first smoothing with the document proportionally to the sentence, and then interpolating with the collection (i.e. the Two Stage Model). Or, alternatively, first interpolating the sentence and the document, and then smoothing with the collection proportional to the sentence length. We shall detail these methods next.

Two-Stage Model (2S):

The two-stage model adopted here is a variant of the well-known two-stage model used for document retrieval [32]. This model is a combination of Dirichlet (DIR) and Jelinek-Mercer (JM) smoothing. Rather than smoothing with the collection model in both stages, we adapt here the model to the characteristics of the SR task and, therefore, the DIR stage uses $p(t|d)$ while the JM stage uses $p(t)$ for smoothing purposes. This is a simple and natural application of the two-stage smoothing for our problem. The formal expression is:

$$p(t|\theta_{s,d}) = (1 - \lambda)\frac{c(t,s) + \mu p(t|d)}{c(s) + \mu} + \lambda p(t) \tag{10}$$

Two-Stage Model, Stages Inverted (2S-I): We propose here a two-stage model where the order in which DIR and JM smoothing methods are applied is inverted:

$$p(t|\theta_{s,d}) = \Big(1 - \beta\Big)\Big((1 - \lambda)p(t|s) + \lambda p(t|d)\Big) + \beta p(t) \tag{11}$$

where $\beta = \frac{\mu}{c(s)+\mu}$. The sentence model is first smoothed using linear interpolation with the document's model. Next, Dirichlet is applied to smooth with the collection model[3]. By smoothing in this way the first stage provides a new estimate of the foreground terms by combining the sentence and the document (through linear interpolation), and then the next stage adjusts the estimates with the background language model proportional to the length of the sentence. By inverting the smoothing methods, different length normalization schemes are applied to the sentence language models. In later sections, we shall analytically and empirically show how the 2S and 2S-I models differ in this respect.

Observe that DIR and JM smoothing can also be included within this framework assuming that $p(q|s,d) = p(q|s)$ and applying DIR or JM to estimate the likelihood. If $p(d|s)$ is uniform, then these models are equivalent to the ones discussed in section 3.1. However, if $p(d|s)$ is not uniform then we get a novel combination of these popular smoothing strategies with the estimation of the importance of sentences in documents. Table 1 summarizes the different proposed models and informs about what configurations are novel (and, therefore, have not been tested in the literature).

[3] As shown in [31], Dirichlet smoothing can be rewritten in a linear interpolation fashion with a proper document-dependent parameter.

Likelihood	**Smoothing**	**Without** $p(d\vert s)$	**With** $p(d\vert s)$
$p(q\vert\theta_s)$	JM	[12, 13, 16]	untested
$p(q\vert\theta_s)$	DIR	[12, 13, 16]	untested
$p(q\vert\theta_{s,d})$	3MM	[18]	untested
$p(q\vert\theta_{s,d})$	2S	untested	untested
$p(q\vert\theta_{s,d})$	2S-I	untested	untested

Table 1 Language Models included in our study. Most of the configurations are novel and have not been tested in the literature.

3.5 Baseline Sentence Retrieval Models

For completeness, we also include the score functions for popular SR models, tfisf [1] and BM25 [22], which we shall employ as baselines. tfisf was adopted in the literature as the state-of-the-art sentence retrieval method [1]. In [16] we demonstrated that it performs similar to tuned BM25. BM25 is a simple adaption of the popular BM25 formula used in document retrieval to the SR case, such that:

$$sim_{\mathrm{BM25}}(s,q) = \sum_{t\in q\cap s} \log \frac{N - sf(t) + 0.5}{sf(t) + 0.5} \cdot \frac{(k_1+1)c(t,s)}{k_1\left((1-b) + b\frac{c(s)}{avsl}\right) + c(t,s)} \cdot \frac{(k_3+1)c(t,q)}{k_3 + c(t,q)} \tag{12}$$

where N is the number of sentences in the collection, $sf(t)$ is the number of sentences that contain t, *avsl* is the average sentence length and k_1, b and k_3 are parameters.

On the other hand, we also used tfisf, which is a state of the art SR baseline. This measure is an adaptation of tf-idf at sentence level:

$$sim_{\mathrm{tfisf}}(s,q) = \sum_{t\in q\cap s} \log(c(t,q)+1)\log(c(t,s)+1)\log\left(\frac{N+1}{0.5+sf(t)}\right) \tag{13}$$

Unlike the BM25 method, this method is parameter-free. Its performance for sentence retrieval has been shown to be comparable to the best performance obtained by BM25 [16, 13].

Besides these models, we also experimented with variants of tfisf and BM25 that support the combination of sentence and contextual statistics. These variants are discussed in Section 4.2.

4 Empirical Study

This section presents the experimental methodology employed to thoroughly evaluate the performance of the proposed models against existing and state of the art models. Particular attention is paid to examining the differences in performance brought about by the inclusion of the local context. Specifically, we hypothesize that:

1. localized smoothing will improve the estimate of the sentence models, resulting in improved effectiveness, and

2. the centrality of a sentence in a document helps to infer the relevance of a sentence, i.e. sentences that briefly summarize a document tend to be more relevant than the rest of sentences in the document.

4.1 Experimental Setup

As previously mentioned, we adopt the SR task as defined in the TREC novelty tracks: given a textual query that represents an information need, a ranked set of documents is supplied and systems have to process this ranking to extract the sentences that are estimated as relevant to the information need. Along with this definition we used all three TREC Novelty Track collections 2002, 2003 and 2004 [5, 28, 27]. Each collection provides the same sentence retrieval task, but under different conditions. In TREC 2002, the track contains 50 topics, extracted from earlier ad hoc tracks. TREC 2003 and TREC 2004 contain also 50 topics each but these were built specifically by assessors for this task. Because in TREC 2002 and TREC 2003 the aim was to find relevant sentences in relevant documents, all the documents of the ranked list of documents in TREC 2002 and TREC 2003 are relevant. In contrast, in TREC 2004 the ranked set of documents contains both relevant and non-relevant documents. In TREC 2002, on average, only the 2.39% of sentences were judged as relevant, while in TREC 2003 and TREC 2004 the number of sentences judged as relevant is higher (39.07% and 15.97%, respectively). All of these collections include complete relevance judgments (i.e. human assessors judged every sentence in the retrieved documents as relevant or non-relevant). By using all three test collection it is possible to assess the robustness of the sentence retrieval methods and thoroughly evaluate their performance.

The baseline methods and the LM models were implemented using the Lemur toolkit[4]. For the experiments, each collection was indexed where standard stop words were removed but stemming was not applied. The corresponding set of topics for each collection was used, where short queries were constructed taking the title field of the TREC Topic. Observe that we use short queries while the teams participating in the TREC novelty tracks were allowed to use the whole topic. This means that the results presented here are not directly comparable to the official TREC results.

For all of our experiments, we report the performance of each method using three standard measures: precision at ten sentences (P@10), mean average precision (MAP) and R-Prec. Observe that the models proposed are recall-oriented in nature, so we would expect to witness gains in terms of MAP, and to some extent R-Prec. This is because the new models are able to promote sentences that do not necessarily match many query terms, but their context matches with some of the query terms. This should enhance the recall of relevant sentences (in particular sentences which may not overlap with the query terms). The usefulness of recall in sentence retrieval can be illustrated using the application scenario presented in the TREC novelty track [5]: where a user is examining the ranked list of documents, and is interested in reviewing all the on-topic sentences but wants to skip through the non-relevant sentences. In this case, navigation could be made more efficient so that they can transverse through all the relevant sentences in all the documents. Whereas in the context of multi-document summarization, having access to all the relevant sentences is also very important. However, the precision oriented measures, P@10 and to some extent R-Prec, also are important for tasks likes query-biased summarization, snippet generation, and question-

[4] www.lemurproject.org

answering. Ideally, the proposed models will be able to enhance both precision and recall based measures, but are likely to gain the largest improvements in terms of recall.

To compare the differences in performance between the different methods, statistical significance tests were applied using the t-test with a 95% confidence level[5].

During the course of our experiments, each method presented in Section 3 was evaluated. Since many of the methods required parameter tuning, we ensured a fair comparison by employing a train-test methodology. Training of each method (except tfisf, which is parameter free) was performed on one of the three TREC novelty datasets. For BM25 we considered the following range of values: k_1=1.0-2.0 (steps of 0.1), b=0.0-1.0 (steps of 0.1) and k_3 was fixed to 0 (the effect of k_3 is negligible with short queries). For the LM methods, λ was set to 0.1-0.9 (steps of 0.1), the range of values of μ (for 2S and 2S-I) was $\{1, 5, 10, 25, 50, 100, 250, 500, 1000, 2500, 5000, 10000\}$ and the range of values for γ (for the 3MM model) was 0.1-0.9 (steps of 0.1). The parameter settings showing best performance were then fixed. These were then used to conduct the remainder of the evaluation, which was performed on the two remaining datasets. We experimented with the three possible training/testing configurations (training with TREC 2002 and testing with TREC 2003 and TREC 2004; training with TREC 2003 and training with TREC 2002 and TREC 2004; and training with TREC 2004 and training with TREC 2002 and TREC 2003) and found the same trends. In the next sections we report and discuss the results achieved by training with TREC 2002 and testing with TREC 2003 and TREC 2004. However, we include the results for the other training/testing configurations in appendix A to further demonstrate that our methods are robust.

Three models may be needed in order to estimate the relevance of a sentence: a sentence model, a local context model (where all the sentences in the document or the surrounding sentences where considered, depending on the type of the smoothing applied) and the background model (which is generated from all the documents in the collection).

When evaluating the LM approaches, we considered different alternatives. On one hand, we study the impact of $p(d|s)$ to specifically study the effect that this extra and novel component has on SR effectiveness. On the other hand, we considered two different contexts: the document (as it was shown in Section 3) and the surrounding sentences (see the below subsection).

4.1.1 Smoothing with Surrounding Sentences

In the previous sections we studied smoothing methods that included $p(t|d)$ within the sentence model, where $p(t|d)$ was estimated using the maximum likelihood estimate of a term in a document. This implies that all terms in the document are related to the sentence. Here, we propose an alternative estimate of $p(t|d)$ which relaxes this assumption, and assumes that only the sentences surrounding the sentence being scored are related. So given a sentence s, the sentences immediately preceding and following s are directly related to it and, therefore, they constitute a closer context to the sentence s. In this way, considering the surrounding sentences only, a more accurate representation of the sentence LM should be obtained, which we anticipate will also lead to improved performance.

In this case, given a sentence s, its context c_s is composed by the previous sentence s_{prev}, the current sentence s and the next sentence in the document s_{next}[6]. Smoothing is performed

[5] The t-test was shown to produce lower error rates than sign and Wilcoxon [23].

[6] If s is the first or the last sentence in the document, then s_{prev} or s_{next} are ignored, respectively.

by using $p(t|c_s)$ instead of $p(t|d)$ in Equations 9, 10 and 11, where $p(t|c_s)$ is the normalized count of t that occurs in s_{prev}, s and s_{next}.

In the next subsection we show the results of this approach and compare them against the results obtained when smoothing with documents instead of surrounding sentences.

4.2 Experimental Results

The first set of experiments tested the effect of localized smoothing *without* $p(d|s)$ (i.e. sentence importance is not considered, all sentences are considered as equally important). Then, we perform a second set of experiments that examines the impact of sentence importance. Finally, we present additional experiments to determine whether or not the baseline models can also be enhanced by including local context.

Influence of localized smoothing: Table 2 reports the parameter setting that optimized performance[7]. Given the TREC 2002 as the training collection, Table 3 shows the performance in the test collections of the methods against the baselines in terms of P@10, MAP and R-Prec. The table shows the performance of models that use either the document as context, or the surrounding sentences. The best performance is presented in bold. Statistically significant differences between a given result and tfisf are marked with an asterisk, and statistically significant differences w.r.t. standard DIR smoothing are marked with a † (DIR provides the LM baseline, which is referred to as LMB). The test results obtained when TREC 2003 and TREC 2004 were used as the training collection are also provided in the Appendix A.

	P@10		***MAP***		***R-Prec***	
BM25	k_1=1.2, b=0, k_3=0		k_1=1.4, b=0, k_3=0		k_1=1.0, b=0, k_3=0	
	p(q|s,d)	***p(q|s,c_s)***	***p(q|s,d)***	***p(q|s,c_s)***	***p(q|s,d)***	***p(q|s,c_s)***
3MM	λ=0.1, γ=0.9	λ=0.7, γ=0.2	λ=0.8, γ=0.1	λ=0.8, γ=0.1	λ=0.1, γ=0.9	λ=0.1, γ=0.4
2S	λ=0.9, μ=250	λ=0.1, μ=500	λ=0.8, μ=5000	λ=0.1, μ=1	λ=0.9, μ=10000	λ=0.1, μ=50
2S-I	λ=0.9, μ=10000	λ=0.8, μ=500	λ=0.9, μ=5000	λ=0.6, μ=500	λ=0.7, μ=1000	λ=0.9, μ=5000
DIR	μ=100		μ=500		μ=250	
JM	λ=0.1		λ=0.1		λ=0.9	

Table 2 Optimal parameter settings in the training collection (TREC 2002) for BM25 and LMs without *p(d|s)*.

In Table 3, where the language models have been trained using TREC 2002, the first prominent result is that the 2S-I smoothing method is the best performing method in terms of MAP and R-Prec. And this novel method is significantly better than the tfisf and DIR baselines, when either surrounding sentences or the entire document is used in the estimate. This is a good result, as it provides a simple and intuitive method that outperforms the long standing benchmark held on these standard test collections. The results in Tables 11 and 13 also show similar improvements.

In terms of P@10, though, the performance of most of the contextually smoothed models is slightly poorer than the baselines. The 2SI method does provide the best performance at P@10 on the TREC 2004 collection, when using the surrounding sentences to smooth

[7] The best parameter settings when smoothing with the surrounding sentences (c_s) are similar.

	$p(q\|s)$				$p(q\|s,d)$			$p(q\|s,c_s)$		
Context	*n/a*				*Document*			*Surrounding Sents.*		
	tfisf	*BM25*	*DIR (LMB)*	*JM*	*3MM*	*2S*	*2S-I*	*3MM*	*2S*	*2S-I*
	TREC 2003									
P@10	.7480	**.7540**†	.6960	.5600	.5020	.5680	.7080	.5200	.4480	.7320
Δ%(tfisf)		*(+0.8)*	*(-7.0)*	*(-25.1)*	*(-32.9)*	*(-24.1)*	*(-5.3)*	*(-30.5)*	*(-40.1)*	*(-2.1)*
Δ%(LMB)	*(+7.5)*	*(+8.3)*		*(-19.5)*	*(-27.9)*	*(-18.4)*	*(+1.7)*	*(-25.3)*	*(-35.6)*	*(+5.2)*
MAP	.3851†	.3852†	.3638	.3474	.3513	.3502	**.4099***†	.3532	.3494	.3893†
Δ%(tfisf)		*(+0.0)*	*(-5.5)*	*(-9.8)*	*(-8.8)*	*(-9.1)*	*(+6.4)*	*(-8.3)*	*(-9.3)*	*(+1.1)*
Δ%(LMB)	*(+5.9)*	*(+5.9)*		*(-4.5)*	*(-3.4)*	*(-3.7)*	*(+12.7)*	*(-2.9)*	*(-4.0)*	*(+7.0)*
R-Prec	.4581†	.4580†	.4457	.4406	.4419	.4459	**.4765***†	.4373	.4374	.4588
Δ%(tfisf)		*(-0.0)*	*(-2.7)*	*(-3.8)*	*(-3.5)*	*(-2.7)*	*(+4.0)*	*(-4.5)*	*(-4.5)*	*(+0.2)*
Δ%(LMB)	*(+2.8)*	*(+2.8)*		*(-1.1)*	*(-0.9)*	*(+0.0)*	*(+6.9)*	*(-1.9)*	*(-1.9)*	*(+2.9)*
	TREC 2004									
P@10	.4300	.4380	.4200	.3580	.2940	.3540	.4300	.3420	.2720	**.4700***
Δ%(tfisf)		*(+1.9)*	*(-2.3)*	*(-16.7)*	*(-31.6)*	*(-17.7)*	*(+0.0)*	*(-20.5)*	*(-36.7)*	*(+9.3)*
Δ%(LMB)	*(+2.4)*	*(+4.3)*		*(-14.8)*	*(-30.0)*	*(-15.7)*	*(+2.4)*	*(-18.6)*	*(-35.2)*	*(+11.9)*
MAP	.2358†	.2368*†	.2240	.2131	.2195	.2203	**.2550***†	.2226	.2204	.2488*†
Δ%(tfisf)		*(+0.4)*	*(-5.0)*	*(-9.6)*	*(-6.9)*	*(-6.6)*	*(+8.1)*	*(-5.6)*	*(-6.5)*	*(+5.5)*
Δ%(LMB)	*(+5.3)*	*(+5.7)*		*(-4.9)*	*(-2.0)*	*(-1.7)*	*(+13.8)*	*(-0.6)*	*(-1.6)*	*(+11.1)*
R-Prec	.3298†	.3300†	.3146	.3010	.3060	.3088	**.3581***†	.3084	.3111	.3418†
Δ%(tfisf)		*(+0.1)*	*(-4.6)*	*(-8.7)*	*(-7.2)*	*(-6.4)*	*(+8.6)*	*(-6.5)*	*(-5.7)*	*(+3.6)*
Δ%(LMB)	*(+4.8)*	*(+4.9)*		*(-4.3)*	*(-2.7)*	*(-1.8)*	*(+13.8)*	*(-2.0)*	*(-1.1)*	*(+8.6)*

Table 3 P@10, MAP and R-Prec in the test collections (TREC 2003 & TREC 2004). Statistically significant differences w.r.t. tfisf are marked with * and w.r.t. LMB are marked with †.

the language models. However, though this is not always significantly different from the baselines.

As previously mentioned, this is perhaps to be expected because the proposed methods are more likely to improve recall. Still, it is very encouraging to see that early precision can also be increased if the smoothing parameters are appropriately set. Recall that we have trained the parameters on a held out test collection, so the performance reported here is not necessarily the best that could be obtained using improved parameter estimation methods. For the remaining of this paper, the focus of the discussion will be on performance with respect to the recall oriented measures, MAP and R-Prec, unless otherwise specified.

In terms of the type of smoothing, i.e. using surrounding sentences or documents, there was no significant differences between the performance obtained with the different estimates. Though, using the complete document was slightly better overall. The other notable point is that the 3MM and 2S localized smoothing methods did not provide improvements to performance. This suggests that the 2S-I smoothing method provides an advantage over these other smoothing methods, which may not necessarily be because of the local information used. We explore the reasons in the next subsection.

Impact of Sentence Importance: In this set of experiments we considered the influence of the local context stemming from the importance of a sentence within a document. Table 4 reports the best settings in the training collections for the proposed LM methods with the sentence importance component. The performance of each method is shown in Table 5 while Figures 1, 2 and 3 provide a bar graph of the P@10, MAP and R-Prec of each method with and without $p(d|s)$. It is clear from these results that the inclusion of the sentence importance results in significantly better retrieval performance for all the LMs over the state of the art method (tfisf). It appears that the impact of the sentence importance dominates the localized smoothing. For instance, given the query “Chinese earthquake”, the 3MM with sentence importance is able to retrieve the following relevant sentence within the top-10 sentences: “Chinese architects from the Ministry of Construction and Hebei Province and the city of Zhangjiakou have begun work on rebuilding earthquake-damaged parts of Hebei and have completed design work on ten types of residential housing for nine villages as models". Nevertheless, this sentence does not appear in the top-10 of the version of 3MM that does not include sentence importance. This is because this sentence summarizes well the document and, therefore, the $p(d|s)$ factor promotes it.

There are not significantly different levels of effectiveness between each of the different smoothing methods. Observe also that the performance of 2S-I is not substantially affected by the sentence importance factor.

	P@10		***MAP***		***R-Prec***	
BM25	k_1=1.2, b=0, k_3=0		k_1=1.4, b=0, k_3=0		k_1=1.0, b=0, k_3=0	
	p(q|s,d)p(d|s)	***p(q|s,c_s)p(d|s)***	***p(q|s,d)p(d|s)***	***p(q|s,c_s)p(d|s)***	***p(q|s,d)p(d|s)***	***p(q|s,c_s)p(d|s)***
3MM	λ=0.3, γ=0.3	λ=0.1, γ=0.1	λ=0.5, γ=0.1	λ=0.6, γ=0.3	λ=0.3, γ=0.2	λ=0.1, γ=0.2
2S	λ=0.1, μ=1	λ=0.2, μ=1000	λ=0.1, μ=1	λ=0.1, μ=5	λ=0.4, μ=10	λ=0.8, μ=1
2S-I	λ=0.1, μ=1	λ=0.2, μ=250	λ=0.1, μ=10	λ=0.4, μ=1	λ=0.1, μ=100	λ=0.1, μ=10
DIR	μ=250		μ=1		μ=25	
JM	λ=0.9		λ=0.1		λ=0.5	

Table 4 Optimal parameter settings in the training collection (TREC 2002) for LMs with *p(d|s)*.

			$p(q\|s)p(d\|s)$		$p(q\|s,d)p(d\|s)$			$p(q\|s,c_s)p(d\|s)$		
Context	*Sentence Only*				*Document*			*Surrounding Sents.*		
	tfisf	*BM25*	*DIR*	*JM*	*3MM*	*2S*	*2S-I*	*3MM*	*2S*	*2S-I*
	TREC 2003									
P@10	.7480†	**.7540**†	.7280	.7320	.7220	.7440	.7360	.7260	.7340	.7280
Δ%(tfisf)		*(+0.8)*	*(-2.7)*	*(-2.1)*	*(-3.5)*	*(-0.5)*	*(-1.6)*	*(-2.9)*	*(-1.9)*	*(-2.7)*
Δ%(LMB)	*(+7.5)*	*(+8.3)*	*(+4.6)*	*(+5.2)*	*(+3.7)*	*(+6.9)*	*(+5.7)*	*(+4.3)*	*(+5.5)*	*(+4.6)*
MAP	.3851†	.3852†	**.4144***†	.4137*†	.4104*†	.4117*†	.4108*†	.4129*†	.4132*†	.4132*†
Δ%(tfisf)		*(+0.0)*	*(+7.6)*	*(+7.4)*	*(+6.6)*	*(+6.9)*	*(+6.7)*	*(+7.2)*	*(+7.3)*	*(+7.3)*
Δ%(LMB)	*(+5.9)*	*(+5.9)*	*(+13.9)*	*(+13.7)*	*(+12.8)*	*(+13.2)*	*(+12.9)*	*(+13.5)*	*(+13.6)*	*(+13.6)*
R-Prec	.4581†	.4580†	**.4802***†	.4800*†	**.4802***†	.4800*†	.4789*†	.4796*†	.4789*†	.4798*†
Δ%(tfisf)		*(-0.0)*	*(+4.8)*	*(+4.8)*	*(+4.8)*	*(+4.8)*	*(+4.5)*	*(+4.7)*	*(+4.5)*	*(+4.7)*
Δ%(LMB)	*(+2.8)*	*(+2.8)*	*(+7.7)*	*(+7.7)*	*(+7.7)*	*(+7.7)*	*(+7.4)*	*(+7.6)*	*(+7.4)*	*(+7.7)*
	TREC 2004									
P@10	.4300	.4380	.4380	**.4420**	.4400	**.4420**	.4380	.4400	.4380	.4380
Δ%(tfisf)		*(+1.9)*	*(+1.9)*	*(+2.8)*	*(+2.3)*	*(+2.8)*	*(+1.9)*	*(+2.3)*	*(+1.9)*	*(+1.9)*
Δ%(LMB)	*(+2.4)*	*(+4.3)*	*(+4.3)*	*(+5.2)*	*(+4.8)*	*(+5.2)*	*(+4.3)*	*(+4.8)*	*(+4.3)*	*(+4.3)*
MAP	.2358†	.2368*†	.2549*†	.2548*†	.2527*†	.2538*†	.2529*†	.2550*†	.2550*†	**.2553***†
Δ%(tfisf)		*(+0.4)*	*(+8.1)*	*(+8.1)*	*(+7.2)*	*(+7.6)*	*(+7.3)*	*(+8.1)*	*(+8.1)*	*(+8.3)*
Δ%(LMB)	*(+5.3)*	*(+5.7)*	*(+13.8)*	*(+13.8)*	*(+12.8)*	*(+13.3)*	*(+12.9)*	*(+13.8)*	*(+13.8)*	*(+14.0)*
R-Prec	.3298†	.3300†	.3522*†	.3520*†	.3504*†	.3513*†	.3508*†	.3510*†	.3520*†	**.3523***†
Δ%(tfisf)		*(+0.1)*	*(+6.8)*	*(+6.7)*	*(+6.3)*	*(+6.5)*	*(+6.4)*	*(+6.4)*	*(+6.7)*	*(+6.8)*
Δ%(LMB)	*(+4.8)*	*(+4.9)*	*(+12.0)*	*(+11.9)*	*(+11.4)*	*(+11.7)*	*(+11.5)*	*(+11.6)*	*(+11.9)*	*(+12.0)*

Table 5 P@10, MAP and R-Prec in the test collections (TREC 2003 & TREC 2004). Statistically significant differences w.r.t. tfisf are marked with * and w.r.t. standard DIR (LMB) are marked with †.

All the models that include $p(d|s)$ are novel, as previous proposals using LMs are solely based on query likelihood estimations. Note also that the three mixture model as proposed in [18] (i.e. without $p(d|s)$) performs worse than the strong and weak baselines (results shown in the 5^{th} column of Table 3).

Incorporating context into the baselines: The baseline models (tfisf and BM25) are context-unaware w.r.t. the local context. Given the findings we have obtained from incorporating local context in the LM framework, it is natural to wonder whether introducing the

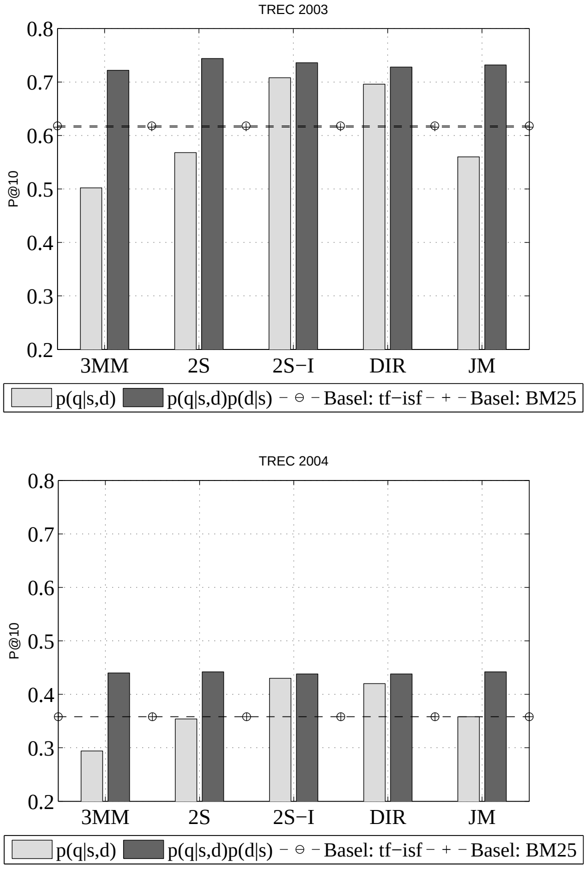

Fig. 1 P@10 in the test collections (TREC 2003 & TREC 2004) of the LMs with and without sentence importance.

local context into the baselines can also improve their performance. First, we present several straight forward adaptions of BM25 and tfisf to include local context, then we compare these variations under the same experimental conditions as above.

A natural solution to introduce document statistics into BM25 [25] is to use the extended version of this model to handle multiple weighted fields, i.e. BM25f [21]. BM25f estimates the relevance of documents considering a document as a set of components. Each of these components may be assigned a specific weight within the document. For our case, a sentence (s) can be considered as an aggregate of the sentence itself and the context containing the sentence (i.e. the document or the surrounding sentences provide local context to the sentence). Given these two components, the BM25f model can be instantiated as follows:

$$sim_{\mathrm{BM25f)}}(s,q) = \sum_{t \in q \cap s} \log \frac{N - sf(t) + 0.5}{sf(t) + 0.5} \cdot \frac{weight(t,s)}{k_1 + weight(t,s)} \cdot \frac{(k_3+1)c(t,q)}{k_3 + c(t,q)} \qquad (14)$$

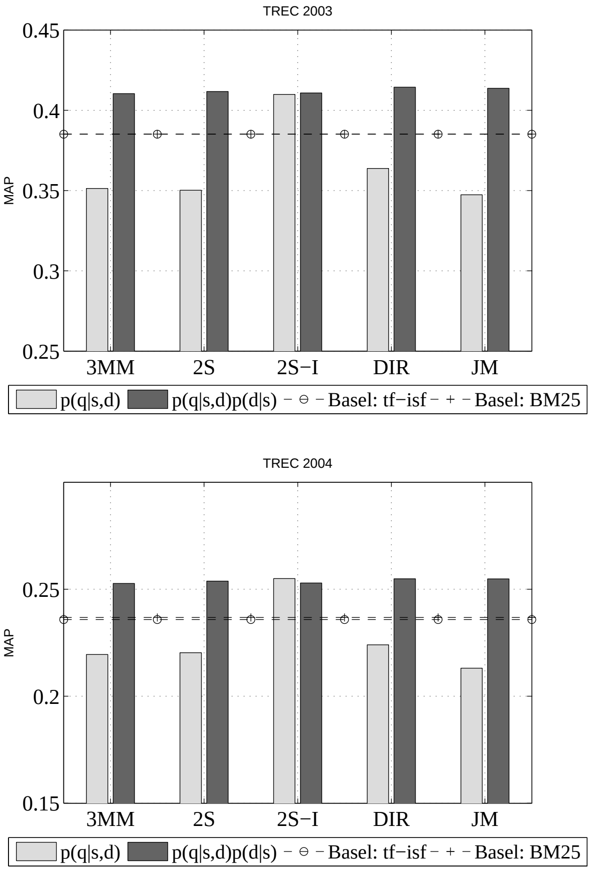

Fig. 2 MAP in the test collections (TREC 2003 & TREC 2004) of the LMs with and without sentence importance.

$$weight(t,s) = \frac{c(t,s)\cdot\alpha}{(1-b_{sen})+b_{sen}\frac{c(s)}{avsl}} + \frac{c(t,context)\cdot(1-\alpha)}{(1-b_{context})+b_{context}\frac{c(context)}{avcl}}$$

where b_{sent} and $b_{context}$ are normalizing constants associated to the field length in s and its context, respectively; α is a boost factor that controls the term frequency mixture between context statistics and sentence statistics; $c(context)$ ($c(s)$) is the number of terms in context (s), $c(t,context)$ is either $c(t,d)$ or $c(t,c_s)$ (depending on whether we apply document-level or surrounding sentences context), and $avcl$ ($avsl$) is the average context (sentence) length in the collection. To reduce the number of parameters to be tuned, $b_{context}$ was fixed to 0.75 (the value usually recommended for document length normalization in BM25 [20]), k_1 was set to the optimal value found with BM25 (Table 2) and k_3 was set again to 0. The remaining parameters, α and b_{sen}, were tuned in the training collection (ranging from 0 to 1 in steps of 0.1).

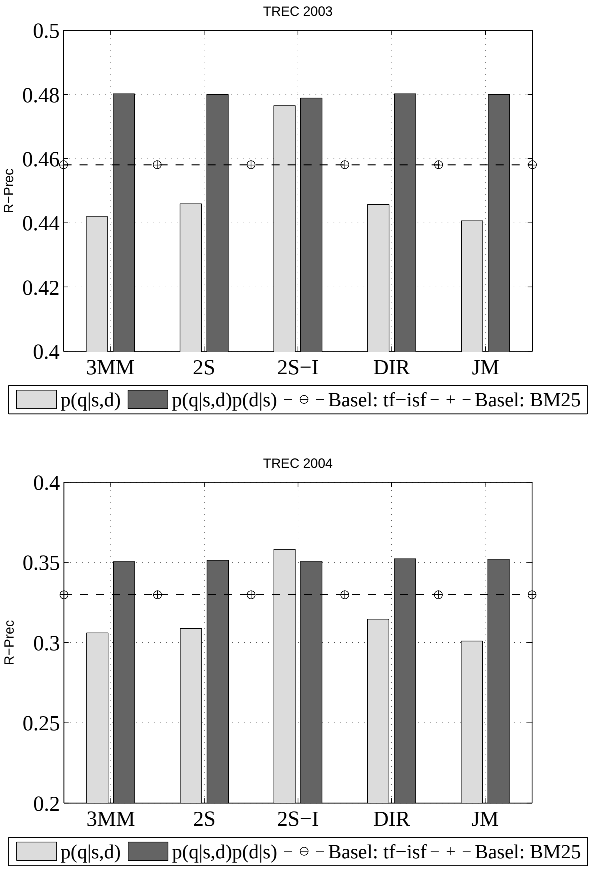

Fig. 3 R-Prec in the test collections (TREC 2003 & TREC 2004) of the LMs with and without sentence importance.

Regarding tfisf, no extensions have been defined to handle local context and, therefore, we defined ad-hoc adjustments to mix context statistics with sentence statistics. We tested the following variants of tfisf:

a) tfmix: c(t,s) is replaced by $\alpha\ c(t,s)+(1-\alpha)c(t,context)$;
b) idfdoc: $sf(t)$ is replaced by $df(t)$ (i.e. idf is computed at the document level rather than at sentence level);
c) tfmix+idfdoc: where both a) and b) were applied.

At training time, only α needs to be tuned (between 0 and 1 in steps of 0.1). Again, TREC 2002 was the training collection and TREC 2003 and TREC 2004 were the test collections. The optimal performance was reached with $b_{sen}=0$ and $\alpha=1$ (BM25f), and $\alpha=1$ (tfisf). This means that these models obtain best performance, when the local context is largely ignored! Tables 6 and 7 report the results achieved in the test collections. Not surprisingly, the variations perform virtually the same as the original models. As a matter of fact, BM25f with $\alpha=1$ (considering either the surrounding sentences or the document as a local context) yields the same SR strategy as BM25. The same happens for tfisf+tfmix ($\alpha=1$) with respect

	BM25	**BM25f**	
		BM25f(d)	**BM25f(c_s)**
		$b_{sen} = 0$, $\alpha = 1$	$b_{sen} = 0$, $\alpha = 1$
	TREC 2003		
P@10	.7540	.7540	.7540
Δ%		*(+0.0)*	*(+0.0)*
MAP	.3852	.3852	.3852
Δ%		*(+0.0)*	*(+0.0)*
R-Prec	.4580	.4580	.4580
Δ%		*(+0.0)*	*(+0.0)*
	TREC 2004		
P@10	.4380	.4380	.4380
Δ%		*(+0.0)*	*(+0.0)*
MAP	.2368	.2368	.2368
Δ%		*(+0.0)*	*(+0.0)*
R-Prec	.3300	.3300	.3300
Δ%		*(+0.0)*	*(+0.0)*

Table 6 Performance of the BM25 and its variations (BM25f) to include context in the test collections (TREC 2003 & TREC 2004).

	tfisf	**idfdoc**	**tfmix**		**tfmix+idfdoc**	
			tfmix(d)	**tfmix(c_s)**	**tfmix+idfdoc(d)**	**tfmix+idfdoc(c_s)**
			$\alpha = 1$	$\alpha = 0.6$	$\alpha = 1$	$\alpha = 0.6$
	TREC 2003					
P@10	.7480	.7540	.7480	.7380	.7540	.7480
Δ%		*(+0.8)*	*(+0.0)*	*(-1.3)*	*(+0.8)*	*(+0.0)*
MAP	.3851	.3906*	.3851	.3843	.3906	.3843
Δ%		*(+1.4)*	*(+0.0)*	*(-0.2)*	*(+1.4)*	*(-0.2)*
R-Prec	.4581	.4613	.4581	.4565	.4613	.4592
Δ%		*(+0.7)*	*(+0.0)*	*(-0.3)*	*(+0.7)*	*(+0.2)*
	TREC 2004					
P@10	.4300	.4360	.4300	.4240	.4360	.4360
Δ%		*(+1.4)*	*(+0.0)*	*(-1.4)*	*(+1.4)*	*(+1.4)*
MAP	.2358	.2363	.2358	.2359	.2363	.2375
Δ%		*(+0.2)*	*(+0.0)*	*(+0.0)*	*(+0.2)*	*(+0.7)*
R-Prec	.3298	.3288	.3298	.3308	.3288	.3270
Δ%		*(-0.3)*	*(+0.0)*	*(+0.3)*	*(-0.3)*	*(-0.8)*

Table 7 Performance of tfisf its variations to include context in the test collections (TREC 2003 & TREC 2004).

to tfisf when the document is considered as the local context. Nevertheless, tfisf+tfmix considering the surrounding sentences ($\alpha = 0.6$) performs worse than tfisf in TREC 2003 and the same as tfisf in TREC 2004. With idfdoc there are some slight variations in performance with respect to the baseline but they are insignificant[8].

While it appears that local context can be useful the model in which it is incorporated determines how successfully this evidence can be used. In the Language Modeling approach, the framework provides a natural and intuitive manner to encode and incorporate the local context through the smoothing process. However, it is unclear how to effectively incorporate the evidence within these other models. We leave this direction for future work, and study more precisely why and how the Language Models are able to capitalize on this additional evidence.

[8] We also tried other values of α on the test collections - and can confirm that when $\alpha = 1$ and $\alpha = 0.6$ the best performance was obtained when the document or the surrounding sentences are considered, respectively.

4.3 Analysis

In this section, we conduct a detailed analysis to understand precisely the reasons behind the differences in effectiveness of the LMs designed. To explain the improvements in performance brought about by the 2S-I model when no sentence importance is used, we derived the retrieval formulas associated to these LMs (similar to that performed in [31,15]). The retrieval formulas in sum-log form are shown in Table 8. Examining the models in this way we can see the differences between each smoothing method. It is interesting to pay attention to the second addend in these formulas. This component incorporates usually some form of length correction. In the DIR and 2S method, this component penalizes long sentences and acts as a length normalization component (which is useful for document retrieval)[9] [14]. In the JM and 3MM methods, this component is independent to the length of the sentence. However, in the 2S-I method, this component *promotes long sentences* because a high $c(s)$ means that β is low making that, overall, the sum is greater (because, usually, $p(t|d) >> p(t)$).

Model	**Retrieval formula**
DIR	$\sum_{t \in s \cap q} c(t,q) \log \left(1 + \frac{c(t,s)}{\mu p(t)}\right) + c(q)\ \log \frac{\mu}{c(s)+\mu}$
JM	$\sum_{t \in s \cap q} c(t,q) \log \left(1 + \frac{(1-\lambda)}{\lambda} \cdot \frac{c(t,s)}{c(s) \cdot p(t)}\right) + c(q) \cdot \log \lambda$
3MM	$\sum_{t \in s \cap q} c(t,q) \log \frac{\lambda p(t \mid s) + \gamma p(t \mid d) + (1-\lambda-\gamma) p(t)}{\gamma p(t \mid d) + (1-\lambda-\gamma) p(t)}$ $+ \sum_{t \in q} c(t,q) \log(\gamma p(t \mid d) + (1-\lambda-\gamma) p(t))$
2S	$\sum_{t \in s \cap q} c(t,q) \log \frac{(1-\lambda) \frac{c(t,s)+\mu p(t \mid d)}{c(s)+\mu} + \lambda p(t)}{(1-\lambda) \frac{\mu p(t \mid d)}{c(s)+\mu} + \lambda p(t)}$ $+ \sum_{t \in q} c(t,q) \log((1-\lambda) \frac{\mu p(t \mid d)}{c(s)+\mu} + \lambda p(t))$
2S-I	$\sum_{t \in s \cap q} c(t,q) \log \frac{(1-\beta)((1-\lambda) p(t \mid s) + \lambda p(t \mid d)) + \beta p(t)}{(1-\beta) \lambda p(t \mid d) + \beta p(t)}$ $+ \sum_{t \in q} c(t,q) \log((1-\beta) \lambda p(t \mid d) + \beta p(t))$ $(\beta = \mu/(c(s)+\mu))$

Table 8 Sum-log retrieval formulas for the SR models based on LMs (without $p(d|s)$).

[9] Note that since older retrieval models such as tf and tf-idf [25] using a vector space model overly favored longer documents, a length correction was required, which penalized longer documents. However, in sentence retrieval it would appear this is not appropriate.

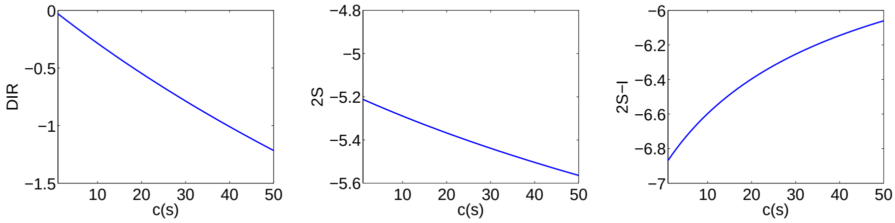

Fig. 4 Effect of non-matching component (length correction) in DIR, 2S and 2S-I against sentence length. The plots show that the score assigned to sentences are adjusted proportionally to the length of the sentence. Note that the 2S-I method favors longer sentences, while the other methods penalize longer sentences.

To illustrate this point further, the Figure 4 shows the behavior of the length correction that the DIR, 2S and 2S-I methods produce with respect to the sentence length. Such correction is given by the second addend of expressions in Table 8. In this example, a query q with three terms (q_A, q_B, q_C) is used, where $c(q_A,q) = c(q_B,q) = c(q_C,q) = 1$, $p(q_A) = 10^{-6}$, $p(q_B) = 10^{-12}$, $p(q_C) = 10^{-3}$, $p(q_A|d) = p(q_B|d) = p(q_C|d) = 10^{-2}$, $\lambda = 0.5$, $\mu = 100$. Then the sentence lengths was varied from 1 to 50 (in steps of 1). Note that in DIR and 2S the correction factor decreases with sentence length, while in 2S-I the value of this factor increases with sentence length. This illustrates graphically that DIR and 2S methods are likely to promote short sentences, while the 2S-I method is likely to promote long sentences.

This seems to indicate that promoting long sentences is a way to achieve better performance, as opposed to using more information. Observe also that the best parameter setting in BM25 fixes b to 0 (Table 2), meaning that sentences are not penalized because of their length. To further support this claim, we analyzed the average length of sentences in these collections and compared it to the average length of relevant sentences. The average sentence length is around 9 terms in all collections, while the average length of relevant sentences is around 14 terms. Furthermore, we analyzed the top 100 sentences retrieved by every model and found that 2S-I yields an average length of 13.71 and 13.66 (TREC 2003 & TREC 2004, respectively), while the other models retrieve shorter sentences on average (e.g. 3MM retrieves sentences whose average length is 12.68 and 12.67, respectively). These statistics suggest that 2S-I is superior to the other models because it promotes longer sentences, and this is required to achieve better performance for the task of sentence retrieval.

Further to this analysis, it is interesting to note that in the estimation of $p(d|s)$ longer sentences will also attract a higher probability. As a matter of fact, in Table 9 and Figure 5 we compare the performance of DIR and JM methods and a variant of them consisting of incorporating a sentence length prior. We show that this variant outperforms significantly their corresponding original versions. However, it does not outperform the 2S-I model and, therefore, the sentence length is not the only component that makes the 2S-I model effective.

Observe that $p(d|s)$, as estimated in section 3.3, is a factor that favors long sentences (because, for the vast majority of the terms in a sentence, $p(t|d) >> p(t)$[10]). This explains why 2S-I does not receive any significant benefits from $p(d|s)$ (as 2S-I already retrieves many long sentences) while the other LM techniques receive significant increases. As a matter of fact, analyzing the top 100 sentences retrieved by every method with $p(d|s)$, we found that the average lengths are quite uniform across models (around 20 terms). This analysis suggests that the local context used indirectly promotes longer sentences, which results in improved retrieval effectiveness.

[10] Recall that $p(\cdot|d)$ and $p(\cdot)$ are both maximum likelihood estimators.

	$p(q\|s)$		$p(q\|s)p(s)$	
	DIR	**JM**	**DIR+len**	**JM+len**
	TREC 2003			
***P*@10**	.6960	.5600	**.7500***	.6120*
	$(\mu = 100)$	$(\lambda = 0.1)$	$(\mu = 250)$	$(\lambda = 0.8)$
***M*AP**	.3638	.3474	**3998***	3730*
	$(\mu = 500)$	$(\lambda = 0.1)$	$(\mu = 50)$	$(\lambda = 0.3)$
***R*-Prec**	.4457	.4406	**4663***	4588*
	$(\mu = 250)$	$(\lambda = 0.9)$	$(\mu = 50)$	$(\lambda = 0.3)$
	TREC 2004			
***P*@10**	.4200	.3580	**.4960***	.3740
	$(\mu = 100)$	$(\lambda = 0.1)$	$(\mu = 250)$	$(\lambda = 0.8)$
***M*AP**	.2240	.2131	**2517***	2298*
	$(\mu = 500)$	$(\lambda = 0.1)$	$(\mu = 50)$	$(\lambda = 0.3)$
***R*-Prec**	.3146	.3010	**3476***	3287*
	$(\mu = 250)$	$(\lambda = 0.9)$	$(\mu = 50)$	$(\lambda = 0.3)$

Table 9 Comparative between DIR and JM against their variants with the sentence length prior (trained with TREC 2002 and tested with TREC 2003 and TREC 2004).

Summary and Discussion: To sum up, the importance of sentences within documents, $p(d|s)$, makes that the performance of the LMs improve significantly beyond existing state of the art. When ignoring $p(d|s)$, 2S-I is the only approach that handles well the retrieval of long sentences with document-level smoothing.

It is quite remarkable that any LM method with $p(d|s)$ is superior to the baselines. This suggests that retrieval methods such as tfisf and BM25 are limited because they are simple adaptations of document retrieval techniques and, therefore, they involve some sort of correction to avoid retrieving many long texts (e.g. b in BM25) but they do not have the opposite tool: some correction to retrieve more long texts. Standard models without length normalization (tfisf or BM25 setting b to 0) have already some tendency towards long pieces of text (because long sentences match more terms) but, given our findings, this is not sufficient to improve the model's performance. However, this also opens the door to future developments, or extensions of current SR models to try to account for this tendency. This will also help to understand whether the important benefits reported here come exclusively from promoting long sentences or, on the contrary, it is the combination of retrieving long sentences and localized smoothing the reason behind such good performance.

5 Conclusions and Future Work

In this paper, we proposed several novel probabilistic LMs to address the SR problem by including the local context. The context provided by the document meant that the estimate of relevance was based on the sentence, the document and the query. As part of the sentence language model, localized smoothing was included to provide a better estimate of the probability of a term in a sentence. The importance of sentences within the document was also included in our models. In a comprehensive set of experiments performed over several TREC test collections, we have compared the proposed models against existing SR models. Our experiments showed that using both forms of local context significantly outperforms the standard LM approach applied to sentence retrieval and the current state of the art sentence retrieval models. This is an important advancement in the development of effective SR methods. More specifically, it was found that:

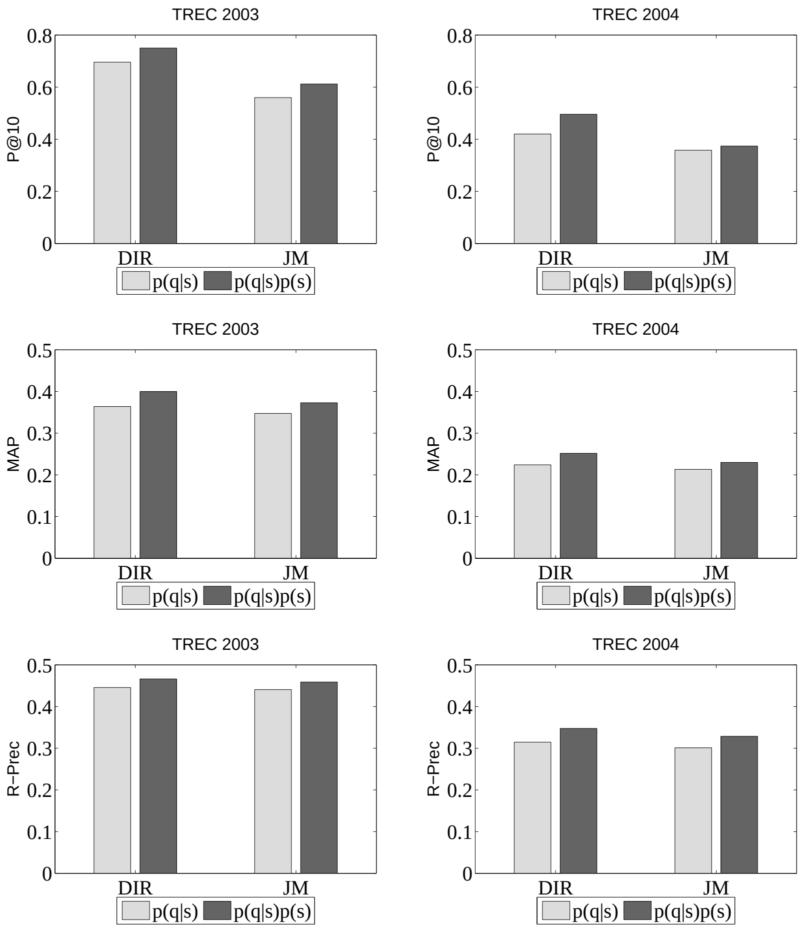

Fig. 5 Comparative between DIR and JM against their variants considering a sentence length prior (trained with TREC 2002 and tested with TREC 2003 and TREC 2004).

- Using localized smoothing (2S-I) improves the performance of the LMs methods (by up to 13.8% improvement in mean average precision (MAP)).
- Including sentence importance significantly improves the performance of all the LM approaches.
- LMs that use local context significantly outperform the current state of the art.

It was also shown that the improvements in the proposed methods were partly due to their tendency to favor longer sentences. This finding demonstrates that the naive application of document retrieval models to other retrieval tasks can lead to non-optimal performance; and warrants the development of sentence retrieval methods which account for the length normalization problem. These findings suggest that further progress in the area of sentence retrieval is possible, and that more sophisticated, and more effective models can be developed by incorporating the local context within the LM framework. This work motivates future research and development on:

(i) developing other methods in a principled fashion to also include local context, i.e. changing the vector representation in tfisf, including a sentence importance factor, or including the local context in the classic Probabilistic Model for IR,

(ii) instead of considering the closest surrounding sentences (previous and next), consider a variable number of surrounding sentences,

(iii) define a four-mixture model that combines the sentence, the local context, the document and the background model,

(iv) the modification of pivoted length normalization, [25] or BM25 to do SR promoting long sentences; or sentence priors for LMs to investigate the length normalization issues,

(v) other estimation methods of the LMs and priors, along with automatic parameter estimation techniques, and

(vi) the application and extension of the Language Modeling Framework to other tasks, such as query-biased summarization or novelty detection.

References

1. James Allan, Courtney Wade, and Alvaro Bolivar. Retrieval and novelty detection at the sentence level. In *Proceedings of the* 26th *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pages 314–321, Toronto, Canada, 2003. ACM.
2. Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. A language modeling framework for expert finding. *Information Processing and Management*, 45(1):1–19, 2009.
3. Kevyn Collins-Thompson, Paul Ogilvie, Yi Zhang, and Jamie Callan. Information filtering, novelty detection and name-page finding. In *Proceedings of the* 11th *Text Retrieval Conference (TREC 2002)*, 2002.
4. Ronald T. Fernández and David E. Losada. Using opinion-based features to boost sentence retrieval. In *Proceedings of the ACM* 18th *Conference on Information and Knowledge Management (CIKM 2009)*, pages 1617–1620, Hong Kong, China, 2009. ACM.
5. Donna Harman. Overview of the TREC 2002 Novelty Track. In *Proceedings of the* 11th *Text Retrieval Conference (TREC 2002)*, pages 46–55, Gaithersburg, USA, 2002.
6. Djoerd Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, January 2001.
7. Nasreen Abdul Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah S. Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. UMass at TREC 2004: Novelty and hard. In *Proceedings of the* 13th *Text Retrieval Conference (TREC 2004)*, volume Special Publication 500-261. National Institute of Standards and Technology (NIST), 2004.
8. Srikanth Kallurkar, Yongmei Shi, R. Scott Cost, Charles K. Nicholas, Akshay Java, Christopher James, Sowjanya Rajavaram, Vishal Shanbhag, Sachin Bhatkar, and Drew Ogle. UMBC at TREC 12. In *Proceedings of the* 12th *Text Retrieval Conference (TREC 2003)*, pages 699–706, 2003.
9. Marcin Kaszkiel and Justin Zobel. Passage retrieval revisted. In *Proceedings of the* 20th *ACM International Conference on Research and Development in Information Retrieval (SIGIR 1997)*, pages 178–185, Philadelphia, USA, 1997. ACM.
10. Marcin Kaszkiel and Justin Zobel. Effective ranking with arbitrary passages. *Journal of The American Society for Informacion Science & Technology*, 52(4):344–364, 2001.
11. Xiaoyan Li and W. Bruce Croft. Novelty detection based on sentence level patterns. In *Proceedings of the* 14th *International Conference on Information and Knowledge Management (CIKM 2005)*, pages 744–751, Bremen, Germany, 2005. ACM.
12. Xiaoyong Liu and W. Bruce Croft. Passage retrieval based on language models. In *Proceedings of the* 11th *International Conference on Information Knowledge and Management (CIKM 2002)*, pages 375–382, Virginia, USA, 2002. ACM.
13. David E. Losada. A study of statistical query expansion strategies for sentence retrieval. In *Proceedings SIGIR 2008 Workshop on Focused Retrieval (Question Answering, Passage Retrieval, Element Retrieval)*, Singapore, 2008. ACM.
14. David E. Losada and Leif Azzopardi. An analysis on document length retrieval trends in language modeling smoothing. *Journal of Information Retrieval*, 11(2):109–138, April 2008.
15. David E. Losada and Leif Azzopardi. Assessing multi-variate Bernoulli models for information retrieval. *ACM Transactions on Information Systems (TOIS)*, 26(3):17:1–17:46, June 2008 2008.

16. David E. Losada and Ronald T. Fernández. Highly frequent terms and sentence retrieval. In *Proceedings of the* 14th *String Processing and Information Retrieval Symposium (SPIRE 2007)*, Lecture Notes in Computer Science, pages 217–228, Santiago de Chile, Chile, 2007. Springer-Verlag.
17. David R. Miller, Tim Leek, and Richard M. Schwartz. A hidden Markov model information retrieval system. In *Proceedings of the* 22th *ACM International Conference on Research and Development in Information Retrieval(SIGIR 1999)*, pages 214–221, Berkeley, US, 1999. ACM.
18. Vanessa G. Murdock. *Aspects of sentence retrieval*. PhD thesis, University of Massachusetts Amherst, September 2006.
19. Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the* 21st *ACM International Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pages 275–281, Melbourne, Australia, 1998. ACM.
20. Stephen Robertson. *In Book TREC: Experiment and Evaluation in Information Retrieval*, chapter How okapi came to TREC, pages 287–299. Digital Libraries and Electronic Publishing. MIT Press, 2005.
21. Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple BM25 extension to multiple weighted fields. In *Proceedings of the* 13th *International Conference on Information and Knowledge Management (CIKM 2004)*, pages 42–49, Washington, USA, 2004. ACM.
22. Stephen E. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC-7: automatic ad hoc, filtering, VCL and interactive track. In *Proceedings of the* 7th *Text Retrieval Conference (TREC 1999)*, pages 253–264, Gaithersburg, USA, 1999.
23. Mark Sanderson and Justin Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *Proceedings of the* 28st *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2005)*, pages 162–169, Salvador, Brazil, 2005. ACM.
24. Luo Si, Rong Jin, Jamie Callan, and Paul Ogilvie. A language modeling framework for resource selection and results merging. In *Proceedings of the* 11th *International Conference on Information and Knowledge Management (CIKM 2002)*, pages 391–397, New York, NY, USA, 2002. ACM.
25. Amit Singhal, Chris Buckley, Mandar Mitra, and Ar Mitra. Pivoted document length normalization. In *Proceedings of the* 19th *ACM International Conference on Research and Development in Information Retrieval (SIGIR 1996)*, pages 21–29. ACM Press, 1996.
26. Mark D. Smucker and James Allan. An investigation of dirichlet prior smoothing's performance advantage. Technical report, University of Massachusetts, Amherst, CIIR, 2005.
27. Ian Soboroff. Overview of the TREC 2004 Novelty Track. In *Proceedings of the* 13th *Text Retrieval Conference (TREC 2004)*, Gaithersburg, USA, 2004.
28. Ian Soboroff and Donna Harman. Overview of the TREC 2003 Novelty Track. In *Proceedings of the* 12th *Text Retrieval Conference (TREC 2003)*, Gaithersburg, USA, 2003.
29. Ryen W. White, Joemon M. Jose, and Ian Ruthven. Using top-ranking sentences to facilitate effective information access. *American Society for Information Science and Technology*, 56(10):1113–1125, 2005.
30. Xiaobing Xue, Jiwoon Jeon, and W. Bruce Croft. Retrieval models for question and answer archives. In *Proceedings of the* 31st *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 475–482, Singapore, 2008. ACM.
31. Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the* 24th *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pages 334–342, New Orleans, USA, 2001. ACM.
32. Chengxiang Zhai and John Lafferty. Two-stage language models for information retrieval. In *Proceedings of the* 25th *ACM International Conference on Research and Development in Information Retrieval(SIGIR 2002)*, pages 49–56. Kluwer Academic Publishers, 2002.
33. Min Zhang, Chuan Lin, Yiqun Liu, Leo Zhao, and Shaoping Ma. THUIR at TREC 2003: Novelty, robust and web. In *Proceedings of the* 12th *Text Retrieval Conference (TREC 2003)*, pages 556–567, Gaithersburg, USA, 2003.

A Appendix

A.1 Localized Smoothing

A.1.1 Training with TREC 2003

	P@10		**MAP**		**R-Prec**	
BM25	k_1=1.1, b=0, k_3=0		k_1=1.4, b=0, k_3=0		k_1=1.1, b=0, k_3=0	
	p(q|s,d)	***p(q|s,c_s)***	***p(q|s,d)***	***p(q|s,c_s)***	***p(q|s,d)***	***p(q|s,c_s)***
3MM	λ=0.9, γ=0.1	λ=0.9, γ=0.1	λ=0.9, γ=0.1	λ=0.9, γ=0.1	λ=0.9, γ=0.1	λ=0.8, γ=0.1
2S	λ=0.4, μ=50	λ=0.2, μ=1	λ=0.6, μ=100	λ=0.1, μ=1	λ=0.9, μ=5000	λ=0.5, μ=1
2S-I	λ=0.3, μ=250	λ=0.3, μ=500	λ=0.8, μ=500	λ=0.9, μ=1000	λ=0.8, μ=1000	λ=0.4, μ=500
DIR	μ=2500		μ=500		μ=100	
JM	λ=0.1		λ=0.1		λ=0.1	

Table 10 Optimal parameter settings in the training collection (TREC 2003) for BM25 and LMs without *p(d|s)*.

	$p(q\|s)$				$p(q\|s,d)$			$p(q\|s,c_s)$		
Context	*n/a*				*Document*			*Surrounding Sents.*		
	tfisf	*BM25*	*DIR (LMB)*	*JM*	*3MM*	*2S*	*2S-I*	*3MM*	*2S*	*2S-I*
					TREC 2002					
P@10	**.2041**	**.2041**†	.1612	.1163	.1122	.1265	.1918†	.1245	.1265	.1755
Δ%(tfisf)		(+0.0)	(-21.0)	(-43.0)	(-45.0)	(-38.0)	(-6.0)	(-39.0)	(-38.0)	(-14.0)
Δ%(LMB)	(+26.6)	(+26.6)		(-27.9)	(-30.4)	(-21.5)	(+19.0)	(-22.8)	(-21.5)	(+8.9)
MAP	.1094†	.1102†	.0937	.0861	.0849	.0938	**.1218***†	.0837	.0916	.1095†
Δ%(tfisf)		(+0.7)	(-14.4)	(.21.3)	(-22.4)	(-14.3)	(+11.3)	(-23.5)	(-16.3)	(+0.1)
Δ%(LMB)	(+16.8)	(+17.6)		(-8.1)	(-9.4)	(+0.1)	(+30.0)	(-10.7)	(-2.2)	(+16.9)
R-Prec	.1659†	.1677†	.1390	.1252	.1385	.1512	**.1841**†	.1367	.1332	.1670†
Δ%(tfisf)		(+1.1)	(-16.2)	(-24.5)	(-16.5)	(-8.9)	(+11.0)	(-17.6)	(-19.7)	(+0.7)
Δ%(LMB)	(+19.4)	(+20.6)		(-9.9)	(-0.4)	(+8.8)	(+32.4)	(-1.7)	(-4.2)	(+20.1)
					TREC 2004					
P@10	.4300	.4380	.4020	.3580	.3560	.3220	.4660*†	.3260	.3420	**.4760***†
Δ%(tfisf)		(+1.9)	(-6.5)	(-16.7)	(-17.2)	(-25.1)	(+8.4)	(-24.2)	(-20.5)	(+10.7)
Δ%(LMB)	(+7.0)	(+9.0)		(-10.9)	(-11.4)	(-19.9)	(+15.9)	(-18.9)	(-14.9)	(+18.4)
MAP	.2358†	.2368*†	.2240	.2131	.2199	.2204	**.2607***†	.2124	.2204	.2496*†
Δ%(tfisf)		(+0.4)	(-5.0)	(-9.6)	(-6.7)	(-6.5)	(+10.6)	(-9.9)	(-6.5)	(+5.9)
Δ%(LMB)	(+5.3)	(+5.7)		(-4.9)	(-1.8)	(-1.6)	(+16.4)	(-5.2)	(-1.6)	(+11.4)
R-Prec	.3298†	.3300†	.3129	.3047	.3105	.3084	**.3552***†	.3174	.3109	.3386†
Δ%(tfisf)		(+0.1)	(-5.1)	(-7.6)	(-5.9)	(-6.5)	(+7.7)	(-3.8)	(-5.7)	(+2.7)
Δ%(LMB)	(+5.4)	(+5.5)		(-2.6)	(-0.8)	(-1.4)	(+13.5)	(+1.4)	(-0.6)	(+8.2)

Table 11 P@10, MAP and R-Prec in the test collections (TREC 2002 & TREC 2004). Statistically significant differences w.r.t. tfisf are marked with * and w.r.t. LMB are marked with †.

A.1.2 Training with TREC 2004

	P@10		***MAP***		***R-Prec***	
BM25	k_1=1.0, b=0, k_3=0		k_1=1.0, b=0, k_3=0		k_1=1.0, b=0, k_3=0	
	p(q|s,d)	***p(q|s,c_s)***	***p(q|s,d)***	***p(q|s,c_s)***	***p(q|s,d)***	***p(q|s,c_s)***
3MM	λ=0.8, γ=0.1	λ=0.8, γ=0.1	λ=0.9, γ=0.1	λ=0.8, γ=0.1	λ=0.7, γ=0.1	λ=0.7, γ=0.1
2S	λ=0.8, μ=10000	λ=0.2, μ=1	λ=0.1, μ=250	λ=0.1, μ=1	λ=0.7, μ=100	λ=0.8, μ=500
2S-I	λ=0.6, μ=250	λ=0.4, μ=500	λ=0.8, μ=100	λ=0.7, μ=500	λ=0.7, μ=100	λ=0.8, μ=500
DIR	μ=250		μ=500		μ=5000	
JM	λ=0.1		λ=0.1		λ=0.1	

Table 12 Optimal parameter settings in the training collection (TREC 2004) for BM25 and LMs without *p(d|s)*.

	p(q|s)				***p(q|s,d)***			***p(q|s,c_s)***		
Context	***n/a***				***Document***			***Surrounding Sents.***		
	tfisf	***BM25***	***DIR (LMB)***	***JM***	***3MM***	***2S***	***2S-I***	***3MM***	***2S***	***2S-I***
	TREC 2002									
P@10	.2041	.2041†	.1633	.1163	.1061	.1531	**.2245**†	.1286	.1265	.1837
Δ%(tfisf)		*(+0.0)*	*(-20.0)*	*(-43.0)*	*(-48.0)*	*(-25.0)*	*(+10.0)*	*(-37.0)*	*(-38.0)*	*(-10.0)*
Δ%(LMB)	*(+25.0)*	*(+25.0)*		*(-28.8)*	*(-35.0)*	*(-6.2)*	*(+37.5)*	*(-21.2)*	*(-22.5)*	*(+12.5)*
MAP	.1094†	.1102†	.0937	.0861	.0849	.0917	**.1200**†	.0919	.0916	.1096†
Δ%(tfisf)		*(+0.7)*	*(-14.4)*	*(-21.3)*	*(-22.4)*	*(-16.2)*	*(+9.7)*	*(-16.0)*	*(-16.3)*	*(+0.2)*
Δ%(LMB)	*(+16.8)*	*(+17.6)*		*(-8.1)*	*(-9.4)*	*(-2.1)*	*(+28.1)*	*(-1.9)*	*(-2.2)*	*(+17.0)*
R-Prec	.1659†	.1677†	.1406	.1252	.1296	.1448	**.1780**†	.1367	.1362	.1682†
Δ%(tfisf)		*(+1.1)*	*(-15.3)*	*(-24.5)*	*(-21.9)*	*(-12.7)*	*(+7.3)*	*(-17.6)*	*(-17.9)*	*(+1.4)*
Δ%(LMB)	*(+18.0)*	*(+19.3)*		*(-11.0)*	*(-7.8)*	*(+3.0)*	*(+26.6)*	*(-2.8)*	*(-3.1)*	*(+19.6)*
	TREC 2003									
P@10	.7480	.7520†	.7140	.5600	.5480	.5800	.7400	.5400	.5320	**.7540**†
Δ%(tfisf)		*(+0.5)*	*(-4.5)*	*(-25.1)*	*(-26.7)*	*(-22.5)*	*(-1.1)*	*(-27.8)*	*(-28.9)*	*(+0.8)*
Δ%(LMB)	*(+4.8)*	*(+5.3)*		*(-21.6)*	*(-23.2)*	*(-18.8)*	*(+3.6)*	*(-24.4)*	*(-25.5)*	*(+5.6)*
MAP	.3851†	.3846†	.3638	.3474	.3555	.3503	**.4098***†	.3532	.3494	.3900†
Δ%(tfisf)		*(-0.1)*	*(-5.5)*	*(-9.8)*	*(-7.7)*	*(-9.0)*	*(+6.4)*	*(-8.3)*	*(-9.3)*	*(+1.3)*
Δ%(LMB)	*(+5.9)*	*(+5.7)*		*(-4.5)*	*(-2.3)*	*(-3.7)*	*(+12.6)*	*(-2.9)*	*(-4.0)*	*(+7.2)*
R-Prec	.4581†	.4580†	.4453	.4416	.4487	.4424	**.4744***†	.4489	.4457	.4611†
Δ%(tfisf)		*(-0.0)*	*(-2.8)*	*(-3.6)*	*(-2.1)*	*(-3.4)*	*(+3.6)*	*(-2.0)*	*(-2.7)*	*(+0.7)*
Δ%(LMB)	*(+2.9)*	*(2.9)*		*(-0.8)*	*(+0.8)*	*(-0.7)*	*(+6.5)*	*(+0.8)*	*(+0.1)*	*(+3.5)*

Table 13 P@10, MAP and R-Prec in the test collections (TREC 2002 & TREC 2003). Statistically significant differences w.r.t. tfisf are marked with * and w.r.t. LMB are marked with †.

A.2 Sentence Importance

A.2.1 Training with TREC 2003

	P@10		***MAP***		***R-Prec***	
BM25	k_1=1.1, b=0, k_3=0		k_1=1.4, b=0, k_3=0		k_1=1.1, b=0, k_3=0	
	p(q|s,d)	***p(q|s,c$_s$)***	***p(q|s,d)***	***p(q|s,c$_s$)***	***p(q|s,d)***	***p(q|s,c$_s$)***
3MM	λ=0.6, γ=0.1	λ=0.7, γ=0.1	λ=0.8, γ=0.1	λ=0.8, γ=0.1	λ=0.1, γ=0.8	λ=0.8, γ=0.1
2S	λ=0.1, μ=1	λ=0.1, μ=1	λ=0.1, μ=1	λ=0.1, μ=1	λ=0.1, μ=250	λ=0.2, μ=1
2S-I	λ=0.1, μ=10	λ=0.1, μ=5	λ=0.1, μ=1	λ=0.1, μ=1	λ=0.8, μ=1	λ=0.7, μ=5
DIR	μ=1		μ=1		μ=10	
JM	λ=0.1		λ=0.1		λ=0.4	

Table 14 Optimal parameter settings in the training collection (TREC 2003) for LMs with *p(d|s)*.

			$p(q\mid s)p(d\mid s)$		$p(q\mid s,d)p(d\mid s)$			$p(q\mid s,c_s)p(d\mid s)$		
Context	***Sentence Only***				***Document***			***Surrounding Sents.***		
	tfisf	***BM25***	***DIR***	***JM***	***3MM***	***2S***	***2S-I***	***3MM***	***2S***	***2S-I***
	TREC 2002									
P@10	.2041†	.2041†	.2429†	.2449†	.2429†	**.2469**†	.2429†	.2449†	.2449†	.2449†
Δ%(tfisf)		*(+0.0)*	*(+19.0)*	*(+20.0)*	*(+19.0)*	*(+21.0)*	*(+19.0)*	*(+20.0)*	*(+20.0)*	*(+20.0)*
Δ%(LMB)	*(+26.6)*	*(+26.6)*	*(+50.7)*	*(+51.9)*	*(+50.7)*	*(+53.2)*	*(+50.7)*	*(+51.9)*	*(+51.9)*	*(+51.9)*
MAP	.1094†	.1102†	**.1349***†	.1347*†	.1333*†	.1344*†	.1329*†	.1342*†	.1343*†	.1347*†
Δ%(tfisf)		*(+0.7)*	*(+23.3)*	*(+23.1)*	*(+21.8)*	*(+22.9)*	*(+21.5)*	*(+22.7)*	*(+22.8)*	*(+23.1)*
Δ%(LMB)	*(+16.8)*	*(+17.6)*	*(+44.0)*	*(+43.8)*	*(+42.3)*	*(+43.4)*	*(+41.8)*	*(+43.2)*	*(+43.3)*	*(+43.8)*
R-Pre	.1659†	.1677†	**.2051***†	.2046*†	.1947†	.1934†	.1943†	.2031*†	.2033*†	.2037*†
Δ%(tfisf)		*(+1.1)*	*(+23.6)*	*(+23.3)*	*(+17.4)*	*(+16.6)*	*(+17.1)*	*(+22.4)*	*(+22.5)*	*(+22.8)*
Δ%(LMB)	*(+19.4)*	*(+20.6)*	*(+47.6)*	*(+47.2)*	*(+40.1)*	*(+39.1)*	*(+39.8)*	*(+46.1)*	*(+46.3)*	*(+46.5)*
	TREC 2004									
P@10	.4300	.4380	.4420	**.4480**	.4400	.4420	.4360	.4460	.4400	.4440
Δ%(tfisf)		*(+1.9)*	*(+2.8)*	*(+4.2)*	*(+2.3)*	*(+2.8)*	*(+1.4)*	*(+3.7)*	*(+2.3)*	*(+3.3)*
Δ%(LMB)	*(+7.0)*	*(+9.0)*	*(+10.0)*	*(+11.4)*	*(+9.5)*	*(+10.0)*	*(+8.5)*	*(+10.9)*	*(+9.5)*	*(+10.4)*
MAP	.2358†	.2368*†	.2549*†	.2548*†	.2531*†	.2538*†	.2532*†	.2550*†	.2551*†	**.2553***†
Δ%(tfisf)		*(+0.4)*	*(+8.1)*	*(+8.1)*	*(+7.3)*	*(+7.6)*	*(+7.4)*	*(+8.1)*	*(+8.2)*	*(+8.3)*
Δ%(LMB)	*(+5.3)*	*(+5.7)*	*(+13.8)*	*(+13.8)*	*(+13.0)*	*(+13.3)*	*(+13.0)*	*(+13.8)*	*(+13.9)*	*(+14.0)*
R-Prec	.3298†	.3300†	.3538*†	.3527*†	.3495†	.3494†	.3496†	**.3545***†	.3536*†	.3538*†
Δ%(tfisf)		*(+0.1)*	*(+7.3)*	*(+6.9)*	*(+6.0)*	*(+5.9)*	*(+6.0)*	*(+7.5)*	*(+7.2)*	*(+7.3)*
Δ%(LMB)	*(+5.4)*	*(+5.5)*	*(+13.1)*	*(+12.7)*	*(+11.7)*	*(+11.7)*	*(+11.7)*	*(+13.3)*	*(+13.0)*	*(+13.1)*

Table 15 P@10, MAP and R-Prec in the test collections (TREC 2002 & TREC 2004). Statistically significant differences w.r.t. tfisf are marked with * and w.r.t. standard DIR (LMB) are marked with †.

A.2.2 Training with TREC 2004

	P@10		***MAP***		***R-Prec***	
BM25	k_1=1.0, b=0, k_3=0		k_1=1.0, b=0, k_3=0		k_1=1.0, b=0, k_3=0	
	p(q|s,d)	***p(q|s,c_s)***	***p(q|s,d)***	***p(q|s,c_s)***	***p(q|s,d)***	***p(q|s,c_s)***
3MM	λ=0.9, γ=0.1	λ=0.4, γ=0.4	λ=0.8, γ=0.1	λ=0.4, γ=0.5	λ=0.6, γ=0.1	λ=0.4, γ=0.5
2S	λ=0.1, μ=1	λ=0.2, μ=25	λ=0.1, μ=1	λ=0.1, μ=1	λ=0.2, μ=1	λ=0.1, μ=25
2S-I	λ=0.2, μ=5	λ=0.3, μ=5	λ=0.1, μ=1	λ=0.1, μ=1	λ=0.4, μ=50	λ=0.4, μ=1
DIR	μ=5		μ=1		μ=1	
JM	λ=0.1		λ=0.1		λ=0.1	

Table 16 Optimal parameter settings in the training collection (TREC 2004) for LMs with *p(d|s)*.

			$p(q\|s)p(d\|s)$		$p(q\|s,d)p(d\|s)$			$p(q\|s,c_s)p(d\|s)$		
Context	*Sentence Only*				*Document*			*Surrounding Sents.*		
	tfisf	*BM25*	*DIR*	*JM*	*3MM*	*2S*	*2S-I*	*3MM*	*2S*	*2S-I*
	TREC 2002									
P@10	.2041†	.2041†	.2449†	.2449†	.1796	**.2469**†	**.2469**†	.2449†	.2449†	.2449†
Δ%(tfisf)		*(+0.0)*	*(+20.0)*	*(+20.0)*	*(-12.0)*	*(+21.0)*	*(+21.0)*	*(+20.0)*	*(+20.0)*	*(+20.0)*
Δ%(LMB)	*(+25.0)*	*(+25.0)*	*(+50.0)*	*(+50.0)*	*(+10.0)*	*(+51.2)*	*(+51.2)*	*(+50.0)*	*(+50.0)*	*(+50.0)*
MAP	.1094†	.1102†	**.1349***†	.1347*†	.1333*†	.1344*†	.1329*†	.1344*†	.1343*†	.1347*†
Δ%(tfisf)		*(+0.7)*	*(+23.3)*	*(+23.1)*	*(+21.8)*	*(+22.9)*	*(+21.5)*	*(+22.9)*	*(+22.8)*	*(+23.1)*
Δ%(LMB)	*(+16.8)*	*(+17.6)*	*(+44.0)*	*(+43.8)*	*(+42.3)*	*(+43.4)*	*(+41.8)*	*(+43.4)*	*(+43.3)*	*(+43.8)*
R-Prec	.1659†	.1677†	**.2041***†	**.2041***†	.2018*†	.2022*†	.2007*†	.2033*†	.2033*†	.2032*†
Δ%(tfisf)		*(+1.1)*	*(+23.0)*	*(+23.0)*	*(+21.6)*	*(+21.9)*	*(+21.0)*	*(+22.5)*	*(+22.5)*	*(+22.5)*
Δ%(LMB)	*(+18.0)*	*(+19.3)*	*(+45.2)*	*(+45.2)*	*(+43.5)*	*(+43.8)*	*(+42.7)*	*(+44.6)*	*(+44.6)*	*(+44.5)*
	TREC 2003									
P@10	.7480†	**.7520**†	.7500	.7480	.6960	.7440	.7360	.7360	.7360	.7420
Δ%(tfisf)		*(+0.5)*	*(+0.3)*	*(+0.0)*	*(-7.0)*	*(-0.5)*	*(-1.6)*	*(-1.6)*	*(-1.6)*	*(-0.8)*
Δ%(LMB)	*(+4.8)*	*(+5.3)*	*(+5.0)*	*(+4.8)*	*(-2.5)*	*(+4.2)*	*(+3.1)*	*(+3.1)*	*(+3.1)*	*(+3.9)*
MAP	.3851†	.3846†	**.4144***†	.4137*†	.4111*†	.4117*†	.4113*†	.4126*†	.4135*†	.4139*†
Δ%(tfisf)		*(-0.1)*	*(+7.6)*	*(+7.4)*	*(+6.8)*	*(+6.9)*	*(+6.8)*	*(+7.1)*	*(+7.4)*	*(+7.5)*
Δ%(LMB)	*(+5.9)*	*(+5.7)*	*(+13.9)*	*(+13.7)*	*(+13.0)*	*(+13.2)*	*(+13.1)*	*(+13.4)*	*(+13.7)*	*(+13.8)*
R-Prec	.4581†	.4580†	.4793*†	.4797*†	.4800*†	**.4805***†	.4795*†	.4791*†	.4803*†	.4800*†
Δ%(tfisf)		*(-0.0)*	*(+4.6)*	*(+4.7)*	*(+4.8)*	*(+4.9)*	*(+4.7)*	*(+4.6)*	*(+4.8)*	*(+4.8)*
Δ%(LMB)	*(+2.9)*	*(+2.9)*	*(+7.6)*	*(+7.7)*	*(+7.8)*	*(+7.9)*	*(+7.7)*	*(+7.6)*	*(+7.9)*	*(+7.8)*

Table 17 P@10, MAP and R-Prec in the test collections (TREC 2003 & TREC 2004). Statistically significant differences w.r.t. tfisf are marked with * and w.r.t. standard DIR (LMB) are marked with †.