Information Retrieval Journal (2020) 23:159–186
https://doi.org/10.1007/s10791-020-09369-x

A passage-based approach to learning to rank documents

Eilon Sheetrit[1] · **Anna Shtok**[2] · **Oren Kurland**[1]

Received: 9 March 2019 / Accepted: 23 February 2020 / Published online: 6 March 2020

Abstract
According to common relevance-judgments regimes, such as TREC's, a document can be deemed relevant to a query even if it contains a very short passage of text with pertinent information. This fact has motivated work on passage-based document retrieval: document ranking methods that induce information from the document's passages. However, the main source of passage-based information utilized was passage-query similarities. In this paper, we address the challenge of utilizing richer sources of passage-based information to improve document retrieval effectiveness. Specifically, we devise a suite of learning-to-rank-based document retrieval methods that utilize an effective ranking of passages produced in response to the query. Some of the methods quantify the ranking of the passages of a document. Others utilize the feature-based representation of the document's passages. Empirical evaluation attests to the clear merits of our methods with respect to highly effective baselines. Our best performing method is based on learning a document ranking function using document-query features and passage-query features of the document's passage most highly ranked; the passage-query features are those used to learn a highly effective passage ranker.

Keywords Document retrieval · Passage retrieval · Learning-to-rank

1 Introduction

The ad hoc retrieval task is ranking documents in a corpus in response to a query by presumed relevance to the information need the query represents. Often, documents are deemed relevant even if they contain only a short passage with pertinent information; e.g., by TREC's relevance judgment regime (Voorhees and Harman 2005). Passages are (relatively short) sequences of text in a document.

✉ Oren Kurland
kurland@ie.technion.ac.il

Eilon Sheetrit
seilon@campus.technion.ac.il

Anna Shtok
annie.shtok@gmail.com

[1] Technion - Israel Institute of Technology, Haifa, Israel

[2] Binyamina, Israel

As a result, there has been a large body of work on *passage-based document retrieval*: utilizing information induced from document passages to rank the documents; e.g., Callan (1994), Wilkinson (1994), Kaszkiel and Zobel (2001), Liu and Croft (2002) and Bendersky and Kurland (2010). The most commonly used passage-based document retrieval methods rank a document by the highest query similarity exhibited by any of its passages (Callan 1994; Wilkinson 1994; Kaszkiel and Zobel 2001; Liu and Croft 2002; Bendersky and Kurland 2010) and by integrating this similarity with the document-query similarity (Callan 1994; Wilkinson 1994; Bendersky and Kurland 2010).

The passage-query (surface level) similarity is one out of many possible estimates for passage relevance. Indeed, various passage-relevance estimates were devised for the task of passage retrieval, a.k.a focused retrieval; e.g., Salton et al. (1993), Jiang and Zhai (2004), Murdock and Croft (2005), Murdock (2006), Metzler and Kanungo (2008), Buffoni et al. (2010), Fernández et al. (2011), Fernández and Losada (2012), Carmel et al. (2013), Keikha et al. (2014b), Chen et al. (2015, 2017), Yang et al. (2016) and Yulianti et al. (2016). That is, passages are ranked in response to a query using passage-relevance estimates. The merits of integrating the estimates using learning-to-rank (LTR) approaches were also demonstrated (Metzler and Kanungo 2008; Buffoni et al. 2010; Chen et al. 2015, 2017; Yang et al. 2016; Yulianti et al. 2016).

Motivated by the (recent) progress in devising effective passage retrieval methods, specifically, using LTR methods, and the fact that the main passage-based information used by most passage-based document retrieval methods is confined to passage-query similarities, we address the following challenge: devising LTR methods for document retrieval that utilize various types of information induced from effective passage ranking. Some of the methods we present are not based on a specific passage retrieval approach used to induce the passage ranking. Others are based on the premise that passages were ranked in response to the query using an LTR method that utilizes passage-based features. A case in point, the most effective LTR-based document retrieval method that we present uses both document-based and passage-based features; the latter are those of the document's passage which is the most highly ranked by an LTR method used to rank passages.

Each of the methods we present can be viewed as a conceptual analog, or generalization, of previously proposed approaches for either (1) passage-based document retrieval, where these approaches do not utilize learning-to-rank or feature-based representations (Callan 1994; Wilkinson 1994; Bendersky and Kurland 2010), or (2) cluster-based document retrieval; i.e., using information induced from clusters of similar documents to improve the effectiveness of document retrieval (Kurland and Domshlak 2008; Raiber and Kurland 2013).

In addition to presenting novel passage-based document retrieval methods, we also propose new features for learning-to-rank passages. These features are query-independent passage relevance priors adapted from work on document retrieval over the Web (Bendersky et al. 2011).

Extensive empirical evaluation shows that our passage-based document retrieval approaches significantly outperform strong baselines. Further analysis demonstrates the importance of (1) utilizing an effective passage ranking, and (2) using information induced from the document's passage that is the most highly ranked. In addition, we demonstrate the merits of using the query-independent passage features we propose for the task of passage retrieval. Specifically, integrating these features with previously proposed ones in a learning-to-rank approach results in passage retrieval performance that transcends the state-of-the-art.

Our main contributions can be summarized as follows:

- We propose a few (mainly learning-to-rank) passage-based document retrieval approaches. Most of these methods are generalization of previously proposed passage-based document retrieval approaches which do not use learning-to-rank or feature-based representations.
- Our proposed methods post state-of-the-art retrieval performance across different collections and different feature sets.
- We demonstrate the effectiveness for passage retrieval of using passage-relevance priors adopted from work on document-relevance priors in Web retrieval.

2 Related work

The line of work most related to ours is on passage-based document retrieval (Hearst and Plaunt 1993; Callan 1994; Mittendorf and Schäuble 1994; Wilkinson 1994; Kaszkiel and Zobel 1997; Denoyer et al. 2001; Kaszkiel and Zobel 2001; Liu and Croft 2002; Bendersky and Kurland 2008; Na et al. 2008; Wang and Si 2008; Wan et al. 2008; Bendersky and Kurland 2010; Krikon et al. 2010; Lang et al. 2010). As already noted, the most commonly used passage-based document retrieval methods are ranking a document by the maximum query-similarity of its passages (Callan 1994; Wilkinson 1994; Kaszkiel and Zobel 1997, 2001; Liu and Croft 2002; Na et al. 2008; Bendersky and Kurland 2010) and by interpolating this similarity with the document-query similarity (Callan 1994; Wilkinson 1994; Na et al. 2008; Bendersky and Kurland 2010). We show that our best-performing methods substantially outperform a highly effective method that integrates document-query and passage-query similarities (Bendersky and Kurland 2010).

In Wang and Si (2008), features based on passage-query similarities were used to learn a document ranker (Wang and Si 2008). The induced ranking was fused with a query-similarity-based document ranking. One of our proposed methods generalizes this approach by using many more passage features, integrating the resultant passage-based document ranking with that produced by learning to rank documents, and applying state-of-the-art learning-to-rank approaches. While this approach is highly effective, it is outperformed by our best performing method.

Recently, Yulianti et al. (2018) presented a method that selects (or generates) a passage from a document in response to a query using information induced from a community question answering system. Then, features of the passage (not necessarily those used for selecting the passage) along with document features are used to represent the document. This approach is reminiscent of our best performing method which uses passage features and document features to represent a document. There are, however, major differences between the two. Our method is not based on an external resource. Furthermore, we utilize passage ranking that is induced using a learning-to-rank approach with passage features while in Yulianti et al. (2018) this is not the case. In addition, the passage features used in our method are the same as those used for ranking passages which is not the case in Yulianti et al. (2018). We demonstrate the merits of using the passage features that are used for (effective) passage ranking to represent a document. We also show the merits of using passage-relevance prior estimates adopted from work on Web retrieval to rank passages. Some of these estimates were used by Yulianti et al. (2018) to rank documents but not passages.

Recently, a neural-network approach was presented for passage-based document retrieval (Fan et al. 2018). Passage-query relevance signals (scores) are estimated using neural-network matching models and then aggregated to yield a document score. A difference with several of our models, in addition to using neural networks rather than a feature-based approach, is that ranking induced over passages from different documents is not utilized. A feature-based learning-to-rank baseline used in this work (Fan et al. 2018) represents a document using its features and the average, maximum and minimum values of query-similarities of its constituent passages. Therefore, this baseline is conceptually reminiscent one of our proposed methods which uses various aggregates of the feature values of document's passages together with the document features to represent documents. We show that there are passage-based features much more effective than passage-query similarities for estimating passage relevance, and accordingly, use aggregates of these features' values to represent documents.

Some passage-based document retrieval methods use query expansion (Liu and Croft 2002; Lang et al. 2010) or inter-passage similarities (Wan et al. 2008; Wang and Si 2008; Krikon et al. 2010). Integrating query expansion and information induced from inter-passage similarities in our approaches is an interesting future direction.

Passage-based document retrieval approaches utilize term proximity information by the virtue of using passages. There are many other approaches for utilizing term proximities (Metzler and Croft 2005, 2007a; Tao and Zhai 2007; Lv and Zhai 2009; Zhao and Yun 2009; Lang et al. 2010; Lv and Zhai 2010; Miao et al. 2012). We show that our best performing method outperforms a state-of-the-art term proximity model: the sequential dependence model from the Markov Random Field framework (Metzler and Croft 2005).

The vast majority of previous work on passage-based document retrieval has focused on using passages marked prior to retrieval time. There are some methods that simultaneously mark passages and use them for retrieval (Mittendorf and Schäuble 1994; Denoyer et al. 2001; Kaszkiel and Zobel 2001). Our methods are not committed to a specific approach of passage markup.

To implement and evaluate our passage-based document retrieval methods, we use a passage ranking method that is based on learning-to-rank. Some of the features we use for passage retrieval are adopted from work on retrieving sentences to create snippets (Metzler and Kanungo 2008) and retrieving sentences (and more generally passages) as answers to non-factoid questions (Keikha et al. 2014b; Chen et al. 2015; Yang et al. 2016). We show that passage retrieval performance can be significantly improved if we also use query-independent passage relevance priors adapted from work on devising document relevance priors for Web retrieval (Bendersky et al. 2011). Query-independent sentence priors different than ours, mainly based on opinion/sentiment analysis, were used in past work on sentence retrieval (Fernández and Losada 2012). More generally, there is a big body of work on retrieving passages; e.g., Salton et al. (1993), Mittendorf and Schäuble (1994), Jiang and Zhai (2004), Carmel et al. (2013), Keikha et al. (2014b), Keikha et al. (2014a), Chen et al. (2017). Our focus is different: we devise methods that utilize passage retrieval to improve document retrieval. Yet, we empirically show that the passage retrieval method we use in our document retrieval methods outperforms state-of-the-art passage retrieval approaches. Still, as already noted, our document retrieval methods are not committed to a specific passage retrieval approach.

3 Retrieval framework

Our goal is to rank documents in corpus $\mathcal{D}$ with respect to query q. We devise document retrieval methods that utilize information induced from document passages. A passage is a sequence of text in a document. We assume that passages were marked in documents using some approach; $g \in d$ indicates that passage g is part of document d. The retrieval methods we present are not dependent on the type of passages used. If $\mathcal{S}$ is a document set, $\mathcal{G}(\mathcal{S})$ denotes the ranked list of all passages of documents in $\mathcal{S}$, where ranking was performed using some passage retrieval method.

Let $\mathcal{D}_{init}$ be an initially retrieved document list produced in response to q by using some retrieval method; e.g., in the experiments reported in Sect. 4 we use standard language-model-based retrieval. Then, a learning-to-rank (LTR) method (Liu 2009) is used to re-rank $\mathcal{D}_{init}$; the resultant ranked list is denoted $\mathcal{D}_{LTR}$. The only assumption we make about the LTR method is that it uses a feature-based vector representation, $\mathbf{v}_{(d,q)}$, for every pair of a document d and the query q.

We devise document ranking methods that re-rank $\mathcal{D}_{LTR}$ using information induced from the ranked list $\mathcal{G}(\mathcal{D}_{LTR})$ of all passages in documents in $\mathcal{D}_{LTR}$.[1] Some of the approaches we present do not depend on the passage ranking method used to produce $\mathcal{G}(\mathcal{D}_{LTR})$. Others are based on the assumption that the ranking is induced using an LTR approach applied to passages; a pair of passage g and query q is represented using the feature vector $\mathbf{v}_{(g,q)}$. The basic premise is that effective passage ranking can be utilized to improve document ranking.

3.1 Passage-based document ranking

We now present five passage-based document retrieval approaches that can be used to re-rank $\mathcal{D}_{LTR}$. These methods are either inspired by, or bear important connections to, existing passage-based and cluster ranking approaches. Cluster ranking methods rank clusters of similar documents by the presumed percentage of relevant documents they contain; e.g., Liu and Croft (2004), Kurland and Krikon (2011) and Raiber and Kurland (2013).

The proposed methods and the different aspects by which they differ are summarized in Table 1. These aspects, as well as the connections and differences between the methods, are discussed below.

3.1.1 A fusion-based approach

The first method we consider is conceptually reminiscent of a commonly used passage-based document retrieval approach. The approach linearly interpolates the document-query similarity score with the highest query similarity score of a passage in the document (Callan 1994; Wilkinson 1994; Bendersky and Kurland 2010).

Here, instead of relying on query similarities, we use the ranking of documents in $\mathcal{D}_{LTR}$ and that of the passages in $\mathcal{G}(\mathcal{D}_{LTR})$ to induce document and passage retrieval scores, respectively. Specifically, we apply the rank-to-score transformation used in the highly effective reciprocal rank fusion method (Cormack et al. 2009). That is, the score assigned to item x, passage or document, with respect to the list $\mathcal{L}$ it is in, $\mathcal{G}(\mathcal{D}_{LTR})$ or $\mathcal{D}_{LTR}$, is:

[1] Note that these passages are also the passages of documents in $\mathcal{D}_{init}$ since $\mathcal{D}_{LTR}$ is a re-rank of $\mathcal{D}_{init}$.

Table 1 Summary of the proposed passage-based document retrieval methods

	RRF	SMPD	JPDs	JPDm	FPD
Document retrieval score					
Fusion of retrieval scores	*				*
LTR		*	*	*	*
Passages that directly affect document retrieval scores					
The document's most highly ranked passage	*		*		*
All of the document's passages		*		*	
Using passages' features to learn a document ranking function?					
Yes			*	*	*
No	*	*			
Assumptions					
The document ranking is induced using an LTR approach		*	*	*	*
The passage ranking is induced using an LTR approach		*	*	*	*

*indicates that a method (column) has (or exhibits) the property (row)

$$Score_{\mathcal{L}}(x) \stackrel{def}{=} \frac{1}{\nu + r_{\mathcal{L}}(x)};$$

$r_{\mathcal{L}}(x)$ is x's rank in $\mathcal{L}$; the top item is at rank 1; ν is a free parameter.

The final retrieval score of document d ($\in \mathcal{D}_{LTR}$) is:

$$Score(d;q) \stackrel{def}{=} \alpha Score_{\mathcal{D}_{LTR}}(d) + (1-\alpha) \max_{g \in d} Score_{\mathcal{G}(\mathcal{D}_{LTR})}(g); \quad (1)$$

α is a free parameter. Thus, d is ranked high if it was originally ranked high in $\mathcal{D}_{LTR}$ and at least one of its passages was ranked high in $\mathcal{G}(\mathcal{D}_{LTR})$.

The method just presented essentially applies the reciprocal rank fusion approach to fuse two rankings of the documents in $\mathcal{D}_{LTR}$ and is therefore denoted **RRF**. The first is the LTR-based ranking of $\mathcal{D}_{LTR}$. That is, documents are ranked using a ranking function learned based on document-only features. The second ranking is based on the highest rank in $\mathcal{G}(\mathcal{D}_{LTR})$ of a document's passage. In other words, the retrieval score of a document with respect to this ranking is based on the reciprocal rank of its passage that is the highest ranked. Note that the method is agnostic to the retrieval methods that were used to produce $\mathcal{D}_{LTR}$ and $\mathcal{G}(\mathcal{D}_{LTR})$; e.g., these need not even be LTR methods. All the method relies on is the ranking of documents and the ranking of passages of these documents.

3.1.2 Utilizing various passage-ranking statistics

The RRF method utilizes only the highest ranked passage of a document to assign its final retrieval score in Eq. 1. The next method, "statistics about multiple passages per document" (**SMPD**), ranks a document by utilizing various *statistics* regarding the ranking of the document's passages in $\mathcal{G}(\mathcal{D}_{LTR})$.

The feature vector used to represent a query-document pair is:

$$\mathbf{v}^{SMPD}_{(d,q)} \stackrel{def}{=} \mathbf{v}_{(d,q)} \oplus \mathbf{v}'_{(g \in d,q)}.$$

$\mathbf{v}^{SMPD}_{(d,q)}$ is the concatenation of $\mathbf{v}_{(d,q)}$: the original feature vector used to learn and apply the ranking function that served to induce $\mathcal{D}_{LTR}$ and $\mathbf{v}'_{(g\in d,q)}$: a vector composed of passage-based estimates. The estimates are the (1) maximum (max), (2) minimum (min), (3) average (avg), and (4) standard deviation (std) of $Score_{\mathcal{G}(\mathcal{D}_{LTR})}(g)$ for $g \in d$; (5) the fraction of passages in d that are among the 50 (top50) and (6) 100 (top100) highest ranked passages in $\mathcal{G}(\mathcal{D}_{LTR})$; and, (7) the number of passages in d (numPsg).

The rationale behind the SMPD method is to augment the original document-query representation with "statistics" about the potential relevance of its passages. The premise is that the relative ranking of passages in $\mathcal{G}(\mathcal{D}_{LTR})$ can attest to their relevance to some extent. While SMPD is based on the fact that $\mathcal{D}_{LTR}$ was indeed produced using an LTR approach, it is not committed to a specific passage ranking method used to produce $\mathcal{G}(\mathcal{D}_{LTR})$.

We note an interesting conceptual connection between SMPD and a cluster-based document retrieval method (Kurland and Domshlak 2008). The method ranks clusters of similar documents using measures that quantify the ranking of their constituent documents in a document ranking. In SMPD, we rank a document using measures that quantify the ranking of its constituent passages.

3.1.3 Joint passage-document representation using a single passage

The next method, "joint passage document with a single passage" (**JPDs**), similarly to the RRF method, uses d's passage g_{max} that is the highest ranked in $\mathcal{G}(\mathcal{D}_{LTR})$. However, JPDs does not rely on g_{max}'s absolute rank in $\mathcal{G}(\mathcal{D}_{LTR})$, but only on the fact that it is the highest ranked among d's passages. JPDs is based on the premise that both $\mathcal{D}_{LTR}$ and $\mathcal{G}(\mathcal{D}_{LTR})$ were produced using LTR methods with feature vectors $\mathbf{v}_{(d,q)}$ and $\mathbf{v}_{(g_{max},q)}$, respectively. These two feature vectors are concatenated, and the resultant feature vector

$$\mathbf{v}^{JPDs}_{(d,q)} \stackrel{def}{=} \mathbf{v}_{(d,q)} \oplus \mathbf{v}_{(g_{max},q)},$$

is used for learning a ranker.

An important principle underlying JPDs is to avoid *metric divergence* (Metzler and Croft 2005). That is, the features used to estimate the relevance of the document's passage that is presumably the most relevant—according to $\mathcal{G}(\mathcal{D}_{LTR})$'s ranking—are used directly, along with document-based features, to learn a document ranking function.

JPDs could be viewed as a conceptual generalization of the approach of smoothing a document language model with that induced from its passage which is the most similar to the query (Bendersky and Kurland 2008). That is, both approaches augment the document representation with information about its passage which is either the most query similar (Bendersky and Kurland 2008) or the most highly ranked using a learning-to-rank approach (JPDs). The difference is unsupervised method (Bendersky and Kurland 2008) versus a supervised method (JPDs), and in accordance, representations (language models vs. feature vectors) and their integration (linear interpolation vs. concatenation).

3.1.4 Joint passage-document representation using multiple passages

The JPDs method uses information induced from a single passage of d to augment the document-query feature-vector representation. We next consider an alternative, "joint passage document with multiple passages"—**JPDm** in short. The document-query representation in JPDm utilizes information induced, potentially, from multiple passages. Specifically, we

define a feature vector, $agg_{g\in d}(\mathbf{v}_{(g,q)})$, based on the same passage features used to represent passages in the LTR method that produced $\mathcal{G}(\mathcal{D}_{LTR})$. Each feature value in $agg_{g\in d}(\mathbf{v}_{(g,q)})$ is the aggregate of the corresponding feature values of all d's passages. The feature vector is then concatenated with the original document-query feature vector

$$\mathbf{v}_{(d,q)}^{JPDm} \stackrel{def}{=} \mathbf{v}_{(d,q)} \oplus agg_{g\in d}(\mathbf{v}_{(g,q)});$$

$\mathbf{v}_{(d,q)}^{JPDm}$ is used for learning a document ranking function. The resultant methods are termed **JPDm-avg**, **JPDm-max** and **JPDm-min** when using the average, maximum and minimum aggregate functions, respectively. We note that JPDm is the only approach we consider which does not use the ranking of passages in $\mathcal{G}(\mathcal{D}_{LTR})$.

It is important to highlight an additional difference between the JPDm and SMPD methods, as both augment the document-query feature vector for learning a document ranking function with information induced from multiple passages in the document. While SMPD uses statistics mainly about the ranking of the document's passages, JPDm utilizes passage-level features which were used to learn a passage ranker. Thus, the empirical comparison between JPDm and SMPD can help to shed some light on the relative merits of using only rank information (SMPD) versus using only feature-based information (JPDm) for multiple passages in the document.

Additional motivation for studying the performance of JPDm is the interesting conceptual connection between JPDm and the ClustMRF cluster ranking model (Raiber and Kurland 2013). In ClustMRF, clusters are ranked with respect to a query using an LTR approach. A cluster-query pair is represented using a feature vector. Some of the features are aggregates of document-query features, where documents are those in the cluster; e.g., document-query similarities and document relevance priors. Similarly, JPDm represents a document using features of passages in the document. Thus, the two approaches are conceptually similar by the virtue of using aggregates of feature values of a "small/short" (doc/passage) entity to represent its ambient entity (document cluster/document).

Finally, we note the important difference between JPDs and JPDm. In JPDs, the passage-based features that are added to the document features represent a single passage; this is the document's most highly ranked passage. In contrast, in JPDm, the passage-based features used to augment the document features do not represent a single passage: these are aggregates, over the document's passages, of feature values used in the passages' feature-vector representations. For example, in JPDm-avg, a single passage-based feature value would be the average feature value—where average is computed over the document's passages—for some feature in the feature-vector representation of the documents' passages.

3.1.5 Two-stage retrieval

To further study the merits of simultaneously using document and passage features to learn a document ranking function as in the JPDs and JPDm methods presented above, we next explore the **FPD** method ("first passage then document").

Table 2 Datasets used for experiments

Corpus	Data	# of docs	Avg doc. length	Queries
ROBUST	Disks 4 and 5-CR	528,155	479	301–450, 601–700
WT10G	WT10g	1,692,096	607	451–550
GOV2	GOV2	25,205,179	930	701–850
ClueWeb	ClueWeb09 (Category B)	50,220,423	807	1–200
INEX	2009 and 2010	2,666,190	552	2009001–2009115, 2010001–2010107
AQUAINT	AQUAINT	1,033,461	436	N1–N100

A *document* ranking function is learned by representing the document-query pair with $\mathbf{v}_{(g_{max},q)}$—the feature vector for the document's passage g_{max} that is the most highly ranked in $\mathcal{G}(\mathcal{D}_{LTR})$. That is, the learned document ranker utilizes only passage-based features. The ranker is then used to re-rank $\mathcal{D}_{LTR}$. The resultant ranking is fused with $\mathcal{D}_{LTR}$'s original ranking using the reciprocal rank approach as in RRF. See Sect. 3.1.1 for further details.[2]

It is important to contrast the FPD and RRF methods. Both fuse the original ranking of $\mathcal{D}_{LTR}$ with a ranking based on utilizing passage-based information. The difference is the type of passage-based information used. While RRF utilizes the rank in $\mathcal{G}(\mathcal{D}_{LTR})$ of the document's most highly ranked passage to directly induce document ranking, FPD utilizes the passage-query feature vector of this passage to learn and apply a document ranker.

We further note that FPD depends on the fact that $\mathcal{G}(\mathcal{D}_{LTR})$ was induced using an LTR approach. In contrast, FPD is not committed to a specific retrieval method used to induce $\mathcal{D}_{LTR}$.

4 Experimental setting

The datasets used for experiments are specified in Table 2. ROBUST, WT10G, GOV2 and ClueWeb are TREC datasets. ROBUST mostly contains newswire documents. WT10G is a small Web corpus. GOV2 is a crawl of the .gov domain. ClueWeb is a large-scale (noisy) Web collection. For ClueWeb we removed from the initial document rankings, described below, documents with a Waterloo's spam classifier score below 50 (Cormack et al. 2011).

The TREC datasets do not have passage-level relevance judgments that are needed for learning a passage-ranking method. Thus, to learn a passage ranker we used the INEX dataset. The learned ranker was utilized by our passage-based document retrieval methods over all datasets. The INEX dataset was used for the focused (passage) retrieval tracks in 2009 and 2010 (Geva et al. 2010; Arvola et al. 2011). It includes relevance judgments for virtually every character in a relevant document; that is, annotators marked the pieces of relevant text in relevant documents. The dataset contains English Wikipedia documents from which we removed all XML tags; i.e., we treated the documents as plaintext. We use this dataset not only for learning a passage ranker, but also for evaluating the effectiveness

[2] Experiments—actual numbers are omitted as they convey no additional insight—showed that simply using the passage-based document ranking without the additional fusion often yields performance (substantially) inferior to that of FPD.

of the learned ranker, as well as evaluating the effectiveness of our passage-based document retrieval methods in addition to the evaluation performed over the TREC datasets.

The passage features we propose are also used for learning and evaluating a passage ranker over the AQUAINT collection which was used for the novelty tracks in TREC 2003 and 2004 (Soboroff and Harman 2003; Soboroff 2004). In these tracks, relevant documents have sentence-level relevance judgments. To perform sentence (passage) retrieval using the queries in both tracks, we follow the experimental setting in the 2003 track and rank the sentences in the set of relevant documents that were provided to participants.

Titles of topics served for queries. (Queries with no relevant documents in the qrels were removed.) The Indri toolkit was used for all experiments.[3] We applied Krovetz stemming to queries, documents (and their passages) and removed stopwords on the INQUERY list only from queries. We used non-overlapping fixed-length windows of 300 terms for passages in our document retrieval methods. Such passages were shown to be effective for passage-based document retrieval (Kaszkiel and Zobel 2001). In Sect. 5 we study the effect of passage length on passage retrieval performance.

Our main experiments are conducted with two learning-to-rank (LTR) methods for ranking documents and passages: LambdaMART (Burges 2010) (LMart in short)[4] or a linear RankSVM (Joachims 2006)[5] (SVM in short). LambdaMART was trained for NDCG@10. In Sect. 5.1.7 we present experimental results for two additional learning-to-rank methods.

We measure the similarity between texts x and y (e.g., a query, a document or a passage) using the minus cross entropy between the unigram language models induced from them:

$$Sim(x, y) \stackrel{def}{=} \exp(-CE(\theta_x^{MLE} \,||\, \theta_y^{Dir})); \tag{2}$$

θ_x^{MLE} is the unsmoothed maximum likelihood estimate induced from x and θ_y^{Dir} is a Dirichlet smoothed language model induced from y (Zhai and Lafferty 2001).

The two-tailed paired t-test with a 95% confidence level was used to determine statistically significant retrieval performance differences. We applied Bonferroni correction for multiple hypothesis testing; i.e., when comparing a method with multiple baselines.

4.1 Document retrieval

We use a standard (unigram) language model approach (**LM**) to retrieve an initial document list $\mathcal{D}_{init}$ of 1000 documents for q: document d is scored by $Sim(q, d)$. We then (re-) rank $\mathcal{D}_{init}$ using an LTR method to obtain $\mathcal{D}_{LTR}$; **init-LTR** denotes this ranking. Since some of the datasets used for evaluation do not have hyperlink and hypertext information, we only use highly effective content-based features. Specifically, the first three features in the document-query feature vector $\mathbf{v}_{(d,q)}$ are those of the sequential dependence model (SDM) from the Markov Random Field (MRF) framework (Metzler and Croft 2005): unigrams, ordered bigrams and unordered bigrams (biterms). SDM is a state-of-the-art term-proximity model. The next three features are the most effective *document relevance priors*

[3] www.lemurproject.org.

[4] Unless otherwise stated, we used the jforests implementation of LambdaMART: https://code.google.com/p/jforests/. In Sect. 5.1.7 we also present the performance results of our best performing method when using the LightGBM implementation of LambdaMART (https://github.com/microsoft/LightGBM).

[5] https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html.

reported in (Bendersky et al. 2011): (1) **SW1** and (2) **SW2** are the fraction of terms in d that are stopwords on the INQUERY list, and the fraction of stopwords on the INQUERY list that appear in d respectively, and (3) the entropy, **Ent**, of the term distribution in d. High presence of stopwords, and high entropy, presumably attests to rich use of language and therefore to content breadth (Bendersky et al. 2011). In Sect. 5.1.8 we also present experimental results when using the MSLR[6] features used in the LETOR datasets.

The set of all passages in documents in $\mathcal{D}_{LTR}$ is ranked to yield $\mathcal{G}(\mathcal{D}_{LTR})$. The same LTR method used to produce $\mathcal{D}_{LTR}$ is used to produce $\mathcal{G}(\mathcal{D}_{LTR})$ with the passage-based features described in Sect. 4.2. Then, $\mathcal{D}_{LTR}$ is re-ranked using the document retrieval methods from Sect. 3 that utilize $\mathcal{G}(\mathcal{D}_{LTR})$. We use MAP and p@10 to evaluate document retrieval performance.

Baselines Recall that $\mathcal{D}_{LTR}$ was attained by re-ranking $\mathcal{D}_{init}$ using an LTR approach; i.e., the set of documents in these two lists is the same. All the baselines we describe and our passage-based document retrieval methods from Sect. 3 are used to rank this document set.

The initial language-model-based ranking of $\mathcal{D}_{init}$, denoted **LM**, is the first baseline. The second is the initial LTR-based ranking of $\mathcal{D}_{LTR}$, init-LTR. MRF's **SDM** with its three features (Metzler and Croft 2005) also serves as a reference comparison. SDM is a special case of the LTR method used to induce $\mathcal{D}_{LTR}$ where document relevance priors are not used.

Another reference comparison is **DocPsg** (Bendersky and Kurland 2010) where document d is scored with $\lambda Sim(q,d) + (1-\lambda)\max_{g \in d} Sim(q,g)$; the value of λ is negatively correlated with d's length which serves as a document homogeneity measure (Bendersky and Kurland 2010). DocPsg is an effective representative of the approach of interpolating document-query and passage-query similarity estimates (Callan 1994; Wilkinson 1994; Bendersky and Kurland 2010).

Additional baseline is the relevance model **RM3** (Abdul-Jaleel et al. 2004). RM3 is a highly effective pseudo-feedback-based query expansion approach. We use RM3 to re-rank $\mathcal{D}_{init}$. It was recently shown—via a system-to-system comparison—that RM3 can outperform advanced neural-network-based document retrieval approaches (Lin 2018).

Finally, we also use as a reference comparison Okapi BM25 (Robertson et al. 1995). Okapi was shown to substantially outperform standard neural network architectures over ROBUST and ClueWeb when using only the queries in these datasets for training (Dehghani et al. 2017), as we do here for the proposed feature-based LTR methods.

4.2 Features for learning to rank passages

All our passage-based document ranking approaches (except for JPDm) utilize a ranking of the documents' passages; i.e., the ranked list $\mathcal{G}(\mathcal{D}_{LTR})$. We now turn to describe the features used for learning a passage ranker. Some of these are novel to this study. The features are estimates of passage g's relevance to the query q. Let d_g denote g's ambient document which we assume is part of a document set S_{doc} retrieved for q. S_{psg} denotes the set of passages of documents in S_{doc}. If S_{doc} is the set of documents in $\mathcal{D}_{LTR}$, the list we aim to re-rank, then S_{psg} is the set of passages in $\mathcal{G}(\mathcal{D}_{LTR})$.

The **PsgQuerySim** feature is the (normalized) passage-query similarity: $\frac{Sim(q,g)}{\sum_{g' \in S_{psg}} Sim(q,g')}$. Since passages are relatively short, the ambient document can provide context in

[6] www.research.microsoft.com/en-us/projects/mslr.

estimating query similarities (cf. Murdock (2006)): **DocQuerySim** is $\frac{Sim(q,d_g)}{\sum_{d' \in S_{doc}} Sim(q,d')}$. Additional document-based features are the maximum, average, and standard deviation of PsgQuerySim for $g' \in d_g$: **MaxPDSim**, **AvgPDSim** and **StdPDSim**, respectively. The longer g is with respect to d_g, the less reliance on document-based query-similarity information is called for (Bendersky and Kurland 2010). Therefore, the ratio between g's and d_g's lengths serves as a query independent feature: **LengthRatio**.

Passages (if exist) that precede (g_{pre}) and follow (g_{follow}) g in d_g provide focused context for g (Fernández et al. 2011). Hence, we use **PsgQuerySimPre** and **PsgQuerySimFollow**: PsgQuerySim for g_{pre} and g_{follow}, respectively. If g is the first or the last passage in the document, we use g's PsgQuerySim for PsgQuerySimPre and PsgQuerySimFollow, respectively.

The next features—the use of which for passage retrieval is novel to this study—are query-independent passage relevance priors. These are adopted from work on document relevance priors in Web search (Bendersky et al. 2011). Specifically, we use the entropy (Ent) and stopwords (SW1, SW2) features described above, but now for passages rather than documents.

The passage independent feature **QueryLength** is the number of unique terms in the query. This feature can potentially help to improve the performance of non-linear rankers (cf., Macdonald et al. (2012)).

The next features are adopted from work on selecting sentences for results' snippets (Metzler and Kanungo 2008). These were also used to retrieve sentences (passages) for questions (Chen et al. 2015; Yang et al. 2016). **ExactMatch** is true if q is a substring of g and false otherwise. **TermOverlap** and **SynonymsOverlap** are the fraction of query terms and their synonyms (determined using Wordnet) in g. **PsgLength** is the number of terms in g after removing stopwords, and **PsgLocation** is g's position (in terms of passages) in d_g over the number of d_g's passages.

We also compare g with q using the following three semantic-similarity measures utilized for sentence-answer retrieval (Yang et al. 2016). (The first two were also used in Chen et al. 2015.) The **ESA** similarity (Gabrilovich and Markovitch 2007) is computed by using, separately, q and the 20 terms in g with the highest TF.IDF values for query likelihood retrieval over the INEX Wikipedia collection. The cosine measure is used to compare the lists of min-max normalized retrieval scores of the top-100 documents.

W2V is the average cosine similarity between any query-term Word2Vec vector and any passage-term Word2Vec vector. We used the 300 dimensional newswire-based Word2Vec vectors from https://code.google.com/p/word2vec/.

Entity is the Jaccard coefficient between the set-based entity representations of q and g. Wikipedia entities (i.e., titles) marked with a confidence level ≥ 0.1 by TagMe (Ferragina and Scaiella 2012) were used.

4.2.1 Evaluating passage retrieval

Most of our passage-based document ranking methods rely on the ranking of document passages. Hence, we also evaluate the effectiveness of the learned passage ranker using the INEX and AQUAINT datasets—this is a focused (passage) retrieval task. For INEX, the set $\mathcal{S}^{init}_{psg}$, of all passages of documents in the language-model-based initially retrieved document list $\mathcal{D}_{init}$, is ranked; the top-1500 passages are evaluated using MAiP and iP[x]: precision at recall level $x \in \{.01, .1\}$ (Geva et al. 2010; Arvola et al. 2011). These evaluation measures were devised for the focused retrieval task where the percentage of relevant

information in a passage is accounted for. For AQUAINT, following the novelty track in 2013 (Soboroff and Harman 2003), we set $\mathcal{D}_{init}$ to be the provided set of relevant documents, and $\mathcal{S}_{psg}^{init}$ is the set of all *sentences* in these documents which are ranked using our passage ranker. The top 1500 ranked sentences are evaluated using MAP and p@10. (The tracks provided sentence-level binary relevance judgments.)

We use the following baselines for passage ranking. The first method, **QSF** ("query-similarity fusion") (Callan 1994; Carmel et al. 2013), scores g by $(1-\lambda)\frac{Sim(q,g)}{\sum_{g' \in \mathcal{S}_{psg}^{init}} Sim(q,g')} + \lambda \frac{Sim(q,d_g)}{\sum_{d' \in \mathcal{D}_{init}} Sim(q,d')}$; λ is a free parameter. The two components of this interpolation are among the features used above for learning a passage ranker.

A tf.idf-based positional model was used for passage retrieval (Carmel et al. 2013). We use a language-model-based positional approach (Lv and Zhai 2009), **PLM**, with a Gaussian kernel, as other methods also utilize language models: g is scored by $\lambda \frac{Sim(q,i_{max}(g))}{\sum_{g' \in \mathcal{S}_{psg}^{init}} Sim(q,i_{max}(g'))} + \beta \frac{Sim(q,g)}{\sum_{g' \in \mathcal{S}_{psg}^{init}} Sim(q,g')} + (1-\lambda-\beta)\frac{Sim(q,d_g)}{\sum_{d' \in \mathcal{D}_{init}} Sim(q,d')}$; $i_{max}(g)$ is the position in g whose Dirichlet induced language model yields the highest query similarity among all positions i in g; λ and β are free parameters. Using PLM as a feature in our passage ranking approach showed no merit.

We adapt the **owpc** method (Buffoni et al. 2010), originally used to rank structured XML elements, as an additional baseline. For compliance with our setting, all features except for those which rely on XML structure are used in the two LTR methods used for all experiments. Most features rely on the query-similarity of the passage and its ambient document; most of the features described above, which we use for learning a passage ranker, were not utilized.

The state-of-the-art LTR-based baseline, **MKS**, utilizes all the features proposed in Yang et al. (2016) for retrieving answer sentences to non-factoid questions. Our passage ranker utilizes some of these features.

The LTR-based approaches, owpc, MKS and our methods, are used to re-rank the top 1500 passages retrieved by QSF which is considered an effective method. Applying LTR methods on an initially retrieved list is common practice (Liu 2009); specifically, the list size, for document retrieval, is often the same as that of the number of documents to be retrieved (e.g., 1000); hence, LTR methods often operate as re-ranking approaches. Similarly, the 1500 threshold used here for passage retrieval corresponds to the standard passage list size used in the focused retrieval track of INEX (Geva et al. 2010; Arvola et al. 2011).

4.3 Additional experimental details

As already noted, we use the INEX dataset to train a passage ranker with the features described in Sect. 4.2. The ranker is also used for passage-based document retrieval over the TREC corpora which lack focused (passage) relevance judgments. To learn a ranker, all passages of documents in the initial language-model-based document list retrieved from INEX, $\mathcal{D}_{init}$, are ranked using the QSF method described in Sect. 4.2.1; thus, $\mathcal{D}_{init}$ serves for the set S_{doc} in Sect. 4.2. The top 1500 passages serve for training. We explored a few binary/graded passage relevance-grade definitions for learning a passage ranker. These use the fraction of relevant characters in a passage, denoted *RFrac*. A bucket-based approach which produces five relevance grades resulted in effective performance of our passage

ranker and the owpc and MKS baselines (see Sect. 4.2.1 for details): 0: $RFrac < .10$; (1) $.10 \leq RFrac < .25$; (2) $.25 \leq RFrac < .50$; (3) $.50 \leq RFrac < .75$; (4) $.75 \leq RFrac$.

To learn a passage ranking function for the sentence retrieval (ranking) task over AQUAINT, we use the sentences' binary relevance judgments as relevance grades.

For the JPDs passage-based document retrieval approach, the DocQuerySim passage feature is not used, as it is the unigram feature of SDM that is used as a document-based feature. For the JPDm-avg and JPDm-max passage-based document retrieval methods, we do not use the passage-query similarity feature PsgQuerySim (see Sect. 4.2) in $agg_{g \in d}(\mathbf{v}_{(g,q)})$ since aggregating this feature value across the passages in the document amounts to the AvgPDSim and MaxPDSim features, respectively, which are already used in $\mathbf{v}_{(g,q)}$.

We used leave-one-out cross validation over queries for training and testing; i.e., each query was used once for test wherein all other queries were used for training. For the LTR methods we randomly split the train set to train (80%) and validation (20%);[7] the latter was used to set the hyper parameters of the LTR methods. For consistency, we use the same train set to set the free-parameter values of the non-LTR baselines (i.e., the validation set is not used for these methods). MAP and MAiP served as the optimization criteria for values of (hyper-) parameters in document and passage retrieval, respectively. We min-max normalized the feature values used in the learning-to-rank methods on a per-query basis.

The Dirichlet smoothing parameter was set to 1000 (Zhai and Lafferty 2001) for the initial language-model-based document retrieval, and to values in $\{500, 1500, 2500\}$ in all other cases unless otherwise specified. The three parameters of MRF 's SDM are set to values in $\{0, 0.1, \ldots, 1\}$. The value of λ in QSF is in $\{0.1, 0.2, \ldots, 0.9\}$. RankSVM 's regularization parameter is set to $\{0.0001, 0.01, 0.1\}$; all other hyper parameters of RankSVM, and those of LambdaMART, are set to default values of the implementations.

For PLM, the value of the steepness parameter of the Gaussian kernel is in $\{50, 100, \ldots, 300\}$; λ and β were set to values in $\{0, 0.2, \ldots, 1\}$ (Lv and Zhai 2009). α (in the RRF and FPD methods from Sect. 3) and ν (in the RRF, SMPD and FPD methods from Sect. 3) are in $\{0, 0.1, \ldots, 1\}$ and $\{0, 30, 60, 90, 100\}$, respectively.

The relevance model RM3 is constructed using unsmoothed maximum likelihood estimates induced from the documents most highly ranked in $\mathcal{D}_{init}$ (Raiber and Kurland 2013).[8] We set the number of documents from which RM3 is constructed, the number of terms and the interpolation parameter that controls the weight of the original query model to values in $\{50, 100\}$, $\{10, 25, 50, 100\}$ and $\{0, 0.1, \ldots, 1\}$, respectively.

For Okapi BM25, the values of the free parameters, k_1 and b, are set to values in $\{0.1, 0.2, \ldots, 4\}$ and $\{0.1, 0.15, \ldots, 1\}$, respectively.

5 Experimental results

In Sect. 5.1 we analyze the performance of our passage-based document retrieval methods described in Sect. 3. As these methods rely on passage ranking, in Sect. 5.2 we analyze the performance of our learning-to-rank-based passage retrieval method.

[7] The only exception was that the passage LTR method applied on TREC corpora was learned using all queries in the INEX dataset.

[8] Not smoothing these language models was shown to yield highly effective RM3 performance (Raiber and Kurland 2013).

Table 3 Main result

	ROBUST		WT10G		GOV2		ClueWeb		INEX	
	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10
LM	.254	.433	.195	.290	.292	.534	.187	.339	.367	.554
DocPsg	.254	.424	.209	.292	.298	.523	.168	.306	.368	.538
SDM	.261	.440	.202	.293	.304	.576	.192	.338	.385	.568
RM3	.281	.443	.196	.303	.325	.571	.198	.361	.390	.568
BM25	.255	.443	.201	.295	.294	.574	.205	.363	.371	.562
init-SVM	.261	.439	.213	.334	.336	.643	.222	.406	.392	.577
init-LMart	.245	.427	.198	.311	.326	.651	.224	.394	.378	.584
JPDs-SVM	$\mathbf{.290}^{lds}_{bi}$	$\mathbf{.480}^{lds}_{rbi}$	$\mathbf{.235}^{lds}_{rbi}$	$\mathbf{.381}^{lds}_{rbi}$	$\mathbf{.350}^{lds}_{rbi}$	$\mathbf{.656}^{lds}_{rb}$	$\mathbf{.246}^{lds}_{rbi}$	$\mathbf{.452}^{lds}_{rbi}$	$\mathbf{.417}^{lds}_{rbi}$	$.589^{ld}$
JPDs-LMart	$\mathbf{.290}^{lds}_{bi}$	$.471^{lds}_{bi}$	$.229^{l}_{i}$	$.378^{lds}_{rbi}$	$.345^{lds}_{bi}$	$.655^{lds}_{rb}$	$.234^{lds}_{rb}$	$.423^{lds}_{rb}$	$.412^{lds}_{rbi}$	$\mathbf{.593}^{ld}$

Comparison between document retrieval baselines and JPDs-LTR which is shown below to be our best performing method. 'l', 'd', 's', 'r', 'b' and 'i' mark statistically significant differences with LM, DocPsg, SDM, RM3, BM25 and init-LTR respectively. Comparisons between LTR-based methods are performed between two methods utilizing the same LTR approach

Boldface: best result per column

5.1 Passage-based document retrieval

5.1.1 Main result

Table 3 presents our main result. We see that in all relevant comparisons (5 datasets × 2 evaluation measures), JPDs, which is shown below to be our best performing approach, substantially outperforms all baselines: LM (unigram language-model-based retrieval), DocPsg (a representative passage-based document retrieval approach), SDM (a state-of-the-art term proximity method), RM3 (a highly effective query expansion approach), BM25 (Okapi BM25) and init-LTR (a learning-to-rank approach that utilizes document-query features). Most improvements are statistically significant. (We applied Bonferroni correction for multiple comparisons.) Refer back to Sect. 4.1 for more details about the baselines.

Recall that JPDs learns a document ranker by utilizing the document-query features used to induce init-LTR and the passage-query features of the document's passage most highly ranked in response to the query. Its clear superiority with respect to the init-LTR methods attest to the merits of the way JPDs leverages passage-based information.

Given the performance superiority in most relevant comparisons of init-SVM and init-LMart to the other baselines, below we use them as reference comparisons. We note that their effectiveness attests to the effectiveness of the document features we use.[9] (See Sect. 4.1 for details regarding the features.)

[9] The finding that init-LMart underperforms init-SVM can be attributed to the fact that LMart is a non-linear ranker while SVM is, and the number of queries used for training is not very large.

Table 4 Comparison of all our passage-based document retrieval methods

	ROBUST		WT10G		GOV2		ClueWeb		INEX	
	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10
init-SVM	.261	.439	.213	.334	.336	.643	.222	.406	.392	.577
init-LMart	.245	.427	.198	.311	.326	.651	.224	.394	.378	.584
JPDs-SVM	.290	**.480**	.235	**.381**	**.350**	.656	**.246**	**.452**	.417	.589
JPDs-LMart	.290	.471	.229	.378	.345	.655	.234	.423	.412	.593
RRF-SVM	$.275^{ij}$	$.462^{ij}$	$.231^{i}$	$.376^{i}$	$.346^{i}$	.639	$.234^{ij}$	$.425^{ij}$	$.408^{ij}$	$.601^{i}$
RRF-LMart	$.281^{ij}$	$.462^{i}$	$.230^{i}$	$.367^{i}$	$.339^{ij}$	.638	$.232^{i}$	$.427^{i}$	$.410^{i}$	.603
SMPD-SVM	$.271^{ij}$	$.455^{ij}$	$.223^{ij}$	$.363^{i}$	$.344^{ij}$	.647	$.233^{ij}$	$.418^{j}$	$.401^{ij}$	$.598^{i}$
SMPD-LMart	$.280^{ij}$	$.460^{i}$	**$.236^{i}$**	$.370^{i}$	$.341^{i}$	.641	$.239^{i}$	$.433^{i}$	$.412^{i}$	.600
JPDm-avg-SVM	$.285^{ij}$	$.465^{ij}$	$.228^{i}$	$.363^{i}$	$.343^{j}$	.639	$.244^{i}$	$.434^{ij}$	$.415^{i}$	$.598^{i}$
JPDm-avg-LMart	$.288^{i}$	$.471^{i}$	$.223^{i}$	$.355^{ij}$	$.342^{i}$	**.663**	$.237^{i}$	$.422^{i}$	$.417^{i}$	.595
JPDm-max-SVM	**$.293^{i}$**	$.476^{i}$	$.235^{i}$	$.374^{i}$	**$.350^{i}$**	.643	$.242^{i}$	$.429^{j}$	**$.420^{i}$**	$.601^{i}$
JPDm-max-LMart	$.289^{i}$	$.468^{i}$	$.228^{i}$	$.363^{i}$	$.349^{i}$	.654	.230	.416	$.416^{i}$	.602
JPDm-min-SVM	$.270^{ij}$	$.451^{j}$	$.233^{i}$	$.342^{j}$	$.334^{j}$	$.630^{j}$	$.236^{i}$	$.430^{ij}$	$.404^{ij}$	.583
JPDm-min-LMart	$.271^{ij}$	$.454^{ij}$	$.220^{i}$	$.338^{j}$	$.337^{ij}$	.640	.230	$.403^{j}$	$.394^{ij}$	.578
FPD-SVM	$.288^{ij}$	$.474^{i}$	$.228^{ij}$	$.372^{i}$	$.348^{i}$	.643	$.238^{ij}$	$.434^{ij}$	$.411^{ij}$	.588
FPD-LMart	$.291^{i}$	$.468^{i}$	$.228^{i}$	$.362^{i}$	$.349^{i}$	.655	$.236^{i}$	$.423^{i}$	$.414^{i}$	**$.609^{i}$**

'i' and 'j' mark statistically significant differences with init-LTR and JPDs-LTR, respectively. Comparisons between LTR-based methods are performed between two methods utilizing the same LTR approach

Boldface: best result per column

Since our methods utilize init-SVM and init-LMart (i.e., the initial list $\mathcal{D}_{LTR}$ or features used to induce it), and using each of the two entails a different experimental setting, we compare X-SVM and X-LMart methods separately.

5.1.2 Comparing all our methods

Table 4 presents the performance comparison of all our proposed passage-based document retrieval methods from Sect. 3. The init-LTR methods serve for reference comparison.

We see in Table 4 that all the proposed methods outperform the init-LTR baselines—often statistically significantly—in the vast majority of relevant comparisons and are never outperformed in a statistically significant manner by a baseline.

JPDs is the most effective approach among those we proposed: its block in the table has the highest number of boldfaced numbers, it outperforms any other approach in most relevance comparisons, and it is never statistically significantly outperformed by other approaches while the reverse often holds. These findings attest to the merits of using the passage-query features of the document's passage most highly ranked together with the document-query features to learn a document ranker.

The JPDm-max approach is the second-best performing. This finding is not entirely surprising: JPDs, which is our best performing method, uses the features of the document's passage most highly ranked while JPDm-max uses per each passage-based feature the maximum value over the document's passages. As could be expected, both JPDm-max and JPDm-avg outperform JPDm-min. That is, using the average or the

Table 5 Comparing variants of JPDs

	ROBUST		WT10G		GOV2		ClueWeb		INEX	
	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10
JPDs-SVM	**.290**	**.480**	.235	**.381**	**.350**	**.656**	**.246**	**.452**	**.417**	.589
JPDs-second-SVM	.277^j	.464^j	**.236**	.363	.341^j	.646	.238^j	.430^j	.414	.594
JPDs-third-SVM	.273^j	.455^j	.232	.363	.338^j	.633^j	.238^j	.434^j	.412	**.598**
JPDs-lowest-SVM	.271^j	.452^j	.231	.347^j	.335^j	.629^j	.226^j	.410^j	.402^j	.577
JPDs-LMart	**.290**	**.471**	**.229**	**.378**	**.345**	**.655**	.234	.423	**.412**	**.593**
JPDs-second-LMart	.280^j	.458	.226	.358	.341	**.655**	**.240**	**.429**	.410	.588
JPDs-third-LMart	.273^j	.455^j	.218	.361	.341	.650	.235	.422	.401^j	.587
JPDs-lowest-LMart	.270^j	.448^j	.219	.336^j	.337^j	.649	.232	.411	.400^j	.581

'j' marks statistically significant differences with JPDs-LTR

Boldface: the best result in a column for each LTR method (SVM or LMart)

maximum of a feature value across the document's passages yields better performance than using the minimal value.

Table 4 also shows that RRF outperforms SMPD in most relevant comparisons when using SVM and the reverse holds when using LMart. However, only the MAP differences between RRF-SVM and SMPD-SVM for ROBUST and INEX are statistically significant. We thus conclude that the most important passage-rank-based information is the rank of a document's most highly ranked passage. (Recall that SMPD uses additional statistics about the ranking of passages of a document.) We attribute these findings to the fact that a document can be deemed relevant even if it contains only a single short relevant passage.

Another observation that we make based on Table 4 is that FPD and JPDs outperform RRF in most relevant comparisons; i.e., using the query-passage features of the passage most highly ranked of a document is more effective than using its rank. Using these features together with document features (JPDs) is more effective than using them separately (FPD) to induce document ranking.

5.1.3 Further analysis of JPDs

We saw above that JPDs is the most effective passage-based document retrieval approach among those we proposed. JPDs uses together the document-query features and the passage-query features of the document's most highly ranked passage so as to learn a document ranking function. In Table 5 we contrast the performance of JPDs with that of its variants that use the passage-query features of the document's second (**JPDs-second**), third (**JPDs-third**) and lowest (**JPDs-lowest**) ranked passages in $\mathcal{G}(\mathcal{D}_{LTR})$.

Table 6 Comparing JPDs with JPD-2 where the features of the document's two most highly ranked passages are used in addition to those of the document

	ROBUST		WT10G		GOV2		ClueWeb		INEX	
	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10
JPDs-SVM	.290	**.480**	**.235**	**.381**	.350	**.656**	.246	**.452**	.417	.589
JPD-2-SVM	**.291**	.473	**.235**	.373	**.351**	.655	**.250**^{j}	**.452**	**.421**	**.601**
JPDs-LMart	.290	.471	.229	**.378**	.345	**.655**	.234	.423	.412	.593
JPD-2-LMart	**.291**	**.473**	**.236**	.375	**.349**	.649	**.235**	**.430**	**.418**	**.605**

'j' marks statistically significant differences with JPDs-LTR

Boldface: the best result in a column for each LTR method (SVM or LMart)

Table 5 shows that the original version, JPDs, outperforms in most relevant comparisons its variants (JPDs-second, JPDs-third and JPDs-lowest). More generally, we see that for almost all datasets, the lower the document's passage, whose passage-query features are used, is ranked, the lower the retrieval performance of the JPDs approach that uses these features.[10] These findings attest to the merits of using the features of the document's most highly ranked passage. They also show the benefit of using information induced from the relative ranking of the document's passages with respect to the query.

5.1.4 Utilizing two passages

Our JPDs method utilizes the features of the document's most highly ranked passage in addition to the document's features. We now consider a variant of JPDs, denoted **JPD-2**, which uses in addition the features of the document's passage which is the second ranked.[11] The feature vectors of the two passages are concatenated with that of the document for learning a document ranker. Table 6 presents the results.

We see in Table 6 that using the two passages (JPD-2-LTR) yields performance that is very similar in most relevant comparisons to that of using a single passage (JPDs-LTR). In only a single case, the performance difference is statistically significant.

5.1.5 The effect of the passage ranker

Our passage-based document retrieval approaches (except for JPDm) utilize information induced from the ranking of passages in the initially retrieved document list, $\mathcal{D}_{init}$. In Table 7 we compare the performance of the approaches when using two different passage ranking methods. The first is the QSF method described in Sect. 4.2.1 which integrates the passage-query similarity value with the query-similarity value of the passage's

[10] We note that the use of the lowest ranked passage did not result in substantial performance decrease due to the length of passages used here: 300; that is, such passages can incorporate a descent amount of information from the entire document, especially in cases of relatively short documents.

[11] To avoid having the same features used for the two passages, the following features were removed from the feature vector of the second ranked passage: DocQuerySim, MaxPDSim, AvgPDSim, StdPDSim and QueryLength.

Table 7 The effect on document ranking effectiveness of the passage ranker: LTR-based (PsgLTR) versus integrating the passage-query similarity with the query-similarity of the passage's ambient document (QSF)

	ROBUST		WT10G		GOV2		ClueWeb		INEX	
	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10
RRF-SVM PsgLTR	**.275***	**.462***	**.231***	**.376***	**.346***	.639	**.234***	**.425***	**.408***	**.601***
RRF-SVM QSF	.261	.442	.215	.324	.336	**.643**	.223	.406	.390	.574
RRF-LMart PsgLTR	**.281***	**.462***	**.230***	**.367***	**.339***	.638	**.232***	**.427***	**.410***	**.603***
RRF-LMart QSF	.257	.442	.204	.318	.326	**.645**	.224	.396	.382	.581
SMPD-SVM PsgLTR	**.271***	**.455***	**.223***	**.363***	**.344**	**.647**	**.233**	**.418**	**.401***	**.598***
SMPD-SVM QSF	.259	.439	.213	.337	.337	.642	.227	.409	.386	.564
SMPD-LMart PsgLTR	**.280***	**.460***	**.236***	**.370***	**.341**	.641	**.239***	**.433***	**.412***	**.600**
SMPD-LMart QSF	.258	.442	.211	.327	.336	**.651**	.223	.407	.389	.579
JPDs-SVM PsgLTR	**.290**	**.480**	**.235**	**.381**	**.350**	**.656**	**.246**	**.452**	**.417**	.589
JPDs-SVM QSF	.288	.474	.233	.373	.347	.647	.245	.441	.414	**.595**
JPDs-LMart PsgLTR	**.290**	.471	.229	**.378**	**.345**	**.655**	**.234**	**.423**	**.412**	.593
JPDs-LMart QSF	.289	**.473**	**.230**	.365	.343	.641	.228	.407	.410	**.597**
FPD-SVM PsgLTR	**.288**	.474	.228	**.372**	**.348***	**.643**	**.238***	**.434**	**.411***	.588
FPD-SVM QSF	.286	**.475**	**.230**	.365	.345	.632	.230	.422	.405	**.591**
FPD-LMart PsgLTR	**.291**	**.468**	**.228**	**.362**	**.349***	**.655***	**.236**	**.423**	**.414**	**.609**
FPD-LMart QSF	.287	**.468**	.225	.361	.344	.631	.233	.417	.411	.605

*marks statistically significant differences between PsgLTR and QSF

Boldface: the best result for evaluation measure in a block

ambient document. The second passage ranking method, **PsgLTR**, was used insofar: SVM or LMart applied with our proposed passage-based features from Sect. 4.2.[12] In Sect. 5.2 we show that the passage-ranking effectiveness of PsgLTR is substantially better than that of QSF.

The message rising from Table 7 is clear: our passage-based document retrieval methods post better performance when using the LTR-based passage ranker than when using the QSF method to rank passages. While most improvements are statistically significant, those for JPDs are not. This finding attests to the robustness of JPDs with respect to the passage ranker used.

5.1.6 Feature analysis for document retrieval

We now present feature analysis for our best performing approach, JPDs. We start by analyzing JPDs-SVM which outperforms JPDs-LMart (see Table 3).

First, we average, per dataset, the weights assigned to features in JPDs-SVM using the different training folds. (Recall that we use leave-one-out cross validation.) Then, the features are ordered in descending order of these averages. Each feature is assigned a score which is the reciprocal of its rank position in the ordered list. Finally, features are ordered

[12] We do not present the comparison for the JPDm approach as it is independent of the passage ranking.

Table 8 Varying the LTR method used in JPDs and in init-LTR

	ROBUST		WT10G		GOV2		ClueWeb		INEX	
	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10
init-SVM	.261	.439	.213	.334	.336	.643	.222	.406	.392	.577
JPDs-SVM	**.290**^{i}	**.480**^{i}	**.235**^{i}	**.381**^{i}	**.350**^{i}	**.656**	**.246**^{i}	**.452**^{i}	**.417**^{i}	**.589**
init-LMart	.245	.427	.198	.311	.326	.651	.224	.394	.378	.584
JPDs-LMart	**.290**^{i}	**.471**^{i}	**.229**^{i}	**.378**^{i}	**.345**^{i}	**.655**	**.234**^{i}	**.423**^{i}	**.412**^{i}	**.593**
init-MART	.258	.439	.203	.305	.332	.640	.216	.403	.381	.565
JPDs-MART	**.285**^{i}	**.462**^{i}	**.211**	**.345**^{i}	**.343**^{i}	**.659**	**.223**	**.415**	**.407**^{i}	**.577**
init-CAscent	.257	.443	.211	.324	.329	**.649**	.212	.406	.377	.586
JPDs-CAscent	**.273**^{i}	**.471**^{i}	**.226**^{i}	**.372**^{i}	**.339**^{i}	.647	**.215**	**.420**	**.382**	**.602**
init-GBM	.259	.442	.199	.316	.327	.629	.221	.393	.380	.563
JPDs-GBM	**.286**^{i}	**.471**^{i}	**.231**^{i}	**.354**^{i}	**.340**^{i}	**.657**^{i}	**.226**	**.404**	**.402**^{i}	**.583**

'i' marks statistically significant difference with init-LTR

Boldface: the best result in a column for each LTR method (SVM, LMart, MART, CAscent or GBM)

by averaging their scores across datasets. The top 10 features[13] according to this analysis are (p and d indicate that the feature is of the passage or the document, respectively): SDM unigrams (d), ESA (p), Entity (p), Ent (d), AvgPDSim (p), MaxPDSim (p), SW2 (d), SDM biterms (d), SynonymsOverlap (p), W2V (p). Thus, both document-based and passage-based features are among the top-5 and top-10. This finding attests to the merits of using both types of features to learn a document ranking function.

We also performed ablation tests for JPDs where we removed one feature at a time. Actual numbers are omitted as they convey no additional insight. We order the features in descending order of the number of cases where their removal resulted in statistically significant performance drop. A case is defined by a dataset and evaluation measure. (We include JPDs-SVM and JPDs-LMart together in this analysis.) We mark the features with (d/p,x): whether the feature is document-based or passage-based (d/p) and the number of cases (x) its removal caused statistically significant performance drop. The ordered list of features is: ESA (p,15), SDM unigrams (d,4), SDM biterms (d,2), SW1 (d,2), Ent (d,1), SW2 (d,1), SDM bigrams (d,1), MaxPDSim (p,1), LengthRatio (p,1), SynonymsOverlap (p,1), pLocation (p,1), Entity (p,1). Thus, as was the case for the SVM-based feature weight analysis from above, ESA which is a passage feature and SDM unigrams which is a document feature are the most important. More generally, the list contains both document and passage features. We note that while the removal of each of the document features resulted in at least one case of statistically significant drop, for quite a few passage features this was not the case; i.e., there is redundancy between the passage features.

We next turn to present feature analysis for the SMPD approach.[14] SMPD uses the same document features as JPDs, but different passage-based features: mainly those which quantify the rank positions of the document's passages in the passage ranking. The results of

[13] JPDs-SVM uses 24 features and JPDs-LMart uses 25 features—the additional feature is the query length which is not useful for a linear ranker.

[14] In this analysis we set ν, the free parameter of SMPD, to a value which is effective across the train folds.

an ablation test, as that performed above, are: max (p,5), SW2 (d,4), SDM unigrams (d,3), SDM biterms (d,2), avg (p,2), numPsg (p,2), Ent (d,1), SW1 (d,1), SDM bigrams (d,1), min (p,1), std (p,1), top50 (p,1). We observe again a mix of document and passage features. The max feature, which quantifies the rank position of the document's most highly ranked passage, is more important than the min and avg features. This finding provides further support to the merits of using information about the highest ranked passage of the document.

5.1.7 LTR methods

Heretofore, we applied our methods using two LTR approaches: RankSVM and LambdaMART. In Table 8, we study the performance of our JPDs method with two additional LTR approaches: MART (Friedman 2001) and coordinate ascent (Metzler and Croft 2007b). MART, known as gradient boosted regression trees, is a non-linear pairwise ranker which combines the outputs obtained by different regression trees. On the other hand, coordinate ascent (CAscent in short) is a linear listwise approach. We used the RankLib.[15] implementations of the MART and CAscent algorithms. In addition, we use the LightGBM.[16] toolkit for an additional implementation of LambdaMART; this serves as a reference comparison to the LambdaMART model presented in Sect. 4 based on the jforests implementation. We refer to LightGBM's LambdaMART model as GBM. CAscent and GBM were trained for NDCG@10.

Table 8 shows that the JPDs method improves over the initial LTR ranking in all relevant comparisons (5 datasets $\times$ 2 evaluation measures $\times$ 5 LTR methods). Most of the improvements for SVM, LMart and GBM are statistically significant while some of the improvements for MART and CAscent are statistically significant.

We also see in Table 8 that in most relevant comparisons, using JPDs with SVM and LMart results in performance that transcends that of its implementations that use MART and CAscent. This finding can be attributed to some extent to the effectiveness of the passage ranking utilized by JPDs. The MAiP effectiveness of the passage ranking induced using MART and CAscent is lower than that attained by using SVM and LMart when using the INEX dataset for passage retrieval evaluation. Specifically, the MAiP performance of SVM, LMart, MART and CAscent is .267, .275, .250 and .259, respectively.

In comparing the performance of the two LambdaMART implementations—LMart (jforests) and GBM (LightGBM)—we observe the following in Table 8. init-GBM outperforms init-LMart in 6 out of 10 relevant comparisons, but we found only the MAP difference for ROBUST to be statistically significant. The opposite holds for JPDs; i.e., using JPDs with LMart outperforms JPDs with GBM in most relevant comparisons, but we found only the MAP difference for INEX to be statistically significant. Although the MAiP of the passage ranker using the INEX dataset is almost the same for LMart (.275) and GBM (.276), the iP[.01]—the interpolated precision at 1% recall point—of LMart is higher than that of GBM; .644 and .632, respectively. Recall that the LTR approaches are used for both the passage ranker and the document ranker.

[15] https://sourceforge.net/p/lemur/wiki/RankLib/

[16] https://github.com/microsoft/LightGBM

Table 9 Using the MSLR (LETOR) document features in comparison to using the features used thusfar for the initial document ranking and in our JPDs method

	GOV2		ClueWeb	
	MAP	p@10	MAP	p@10
init-SVM	.336	.643	.222	.406
init-LMart	.326	.651	.224	.394
JPDs-SVM	**.350**^{i}	**.656**	**.246**^{i}	**.452**^{i}
JPDs-LMart	**.345**^{i}	**.655**	**.234**^{i}	**.423**^{i}
init-MSLR-SVM	.323	.595	.251	.437
init-MSLR-LMart	.315	.599	.241	.428
JPDs-MSLR-SVM	**.353**^{m}	**.634**^{m}	**.264**^{m}	**.452**^{m}
JPDs-MSLR-LMart	**.342**^{m}	**.633**^{m}	**.244**	**.437**

'i' and 'm' mark statistically significant differences with init-LTR and init-MSLR-LTR, respectively

Boldface: the best result in a column, per block of either the original features (first block) or the MSLR features (second block), for each LTR method (SVM or LMart)

Table 10 Passage retrieval over INEX with passages of length 300, 150 and 50. LM is standard language-model-based document retrieval (i.e., documents serve for passages)

	INEX								
	Psg300			Psg150			Psg50		
	MAiP	iP[.01]	iP[.1]	MAiP	iP[.01]	iP[.1]	MAiP	iP[.01]	iP[.1]
LM	.256	.523	.449	.256	.523	.449	.256	.523	.449
QSF	.248	.577	.453	.234	.575	.455	.209	.581	.449
PLM	.253	.586	.472	.240	.596	.471	**.215**	.605	**.469**
owpc-SVM	.242	.577	.440	.229	.569	.438	.202	.570	.431
owpc-LMart	.255	.578	.460	.240	.566	.450	.208	.577	.443
MKS-SVM	.247	.593	.468	.235	.602	.459	.199	.626	.457
MKS-LMart	.262	.620	.479	.241	.629	.479	.200	.644	.459
PsgLTR-SVM	$.267_{ok}$	$.637^{lf}_{o}$	$.487_{o}$	$\mathbf{.253_{o}}$	$\mathbf{.662^{lfm}_{ok}}$	$.492_{o}$	$.213^{l}$	$\mathbf{.647^{l}_{o}}$	.467
PsgLTR-LMart	$\mathbf{.275^{fm}_{o}}$	$\mathbf{.644^{lfm}_{o}}$	**.496**	**.253**	$.650^{lf}_{o}$	$\mathbf{.494_{o}}$	$.209^{l}$	$.634^{l}$	.454

Statistically significant differences with LM, QSF and PLM are marked with 'l', 'f' and 'm', respectively. 'o' and 'k' mark a statistically significant difference between PsgLTR-X and owpc-X and between PsgLTR-X and MKS-X, respectively

Boldface: the best result in a column

5.1.8 Using LETOR features

We have used the document features described in Sect. 4.1. This practice resulted in highly effective document ranking performance as exhibited by the init-LTR baselines as well as our methods. We now turn to explore the performance of our methods with a much larger set of document(-query) features. Specifically, we use the MSLR[17] features

[17] www.research.microsoft.com/en-us/projects/mslr.

Table 11 Sentence retrieval over AQUAINT

	AQUAINT	
	MAP	p@10
QSF	.471	.624
PLM	.518	.669
owpc-SVM	.579	.701
owpc-LMart	.589	**.716**
MKS-SVM	.569	.664
MKS-LMart	.585	.701
PsgLTR-SVM	$.602^{fm}_{ok}$	$.713^{f}_{k}$
PsgLTR-LMart	$\mathbf{.606^{fm}_{ok}}$	$.710^{f}$

Statistically significant differences with QSF and PLM are marked with 'f' and 'm', respectively. 'o' and 'k' mark a statistically significant difference between PsgLTR-X and owpc-X and between PsgLTR-X and MKS-X, respectively

Boldface: the best result in a column

from the LETOR datasets for retrieval over the GOV2 and ClueWeb collections with the queries specified in Table 2. We used all MSLR features except for the Outlink number, SiteRank, QualityScore, QualityScore2, Query-url click count, url click count, and url dwell time. In addition to the MSLR features, we also use here the highly effective query-independent document quality measures used above: the fraction of terms in the document that are stopwords, the fraction of stopwords that appear in the document, and the entropy of the term distribution in the document. The stopword list used for the two stopword features is composed of the collection's 100 most frequent alphanumeric terms (Ntoulas et al. 2006; Raifer et al. 2017). For ClueWeb we also used the spam score assigned to a document by the Waterloo spam classifier and the PageRank score. All together, we used, at the document level, 149 features for GOV2 and 151 features for ClueWeb.

The results are presented in Table 9. We first see that in terms of the initial ranking, the MSLR features are more effective than those we used above for ClueWeb, but the reverse holds for GOV2. (This could potentially be attributed to the fact that for GOV2 there are fewer queries than for ClueWeb.) We further see in Table 9 that our JPDs method is also effective with the MSLR features. It always outperforms the initial ranking; in most relevant comparisons, the improvements are statistically significant.

5.2 Passage retrieval

Heretofore, we have focused on the document retrieval task. Our passage-based document retrieval methods utilize a ranking of passages induced using our proposed passage retrieval approach. (See Sect. 4.2 for details.) We now turn to compare the performance of our passage ranker with that of the passage retrieval baselines described in Sect. 4.2.1.

Table 10 presents the performance numbers of the passage retrieval methods for the INEX collection. We see that our LTR methods, PsgLTR-SVM and PsgLTR-LMart, outperform all other passage retrieval methods in most relevant comparisons (3 passage lengths × 3 evaluation measures) with many of the improvements being statistically

significant. We note that the MKS baseline (Yang et al. 2016) was shown to yield state-of-the-art passage retrieval performance.

Table 11 presents the effectiveness of our passage retrieval approach, PsgLTR, in ranking sentences in the AQUAINT collection. We see that PsgLTR-SVM and PsgLTR–LMart statistically significantly outperform all other passage retrieval methods in terms of MAP. In the single case where our methods are outperformed by another method (owpc-LMart) in terms of p@10, the performance differences are not statistically significant.

The findings presented above for focused (passage) retrieval over INEX, and sentence retrieval over AQUAINT, attest to the fact that our passage ranker posts state-of-the-art passage retrieval performance.

5.2.1 Feature analysis for passage retrieval

We first use the SVM-based feature analysis, as was performed above for document retrieval, to analyze the relative importance of features used in our passage retrieval approach (PsgLTR-SVM). For INEX, we consider each of the three passage lengths as a different experimental setting. The top 10 features for INEX are: ESA, SW1, MaxPDSim, Entity, StdPDSim, SW2, Ent, DocQuerySim, AvgPDSim and SynonymsOverlap. For AQUAINT, the top-10 features are: Ent, SW1, ESA, LengthRatio, TermOverlap, AvgPDSim, PsgQuerySimPre, SynonymsOverlap, PsgQuerySimFollow, PsgLength. Recall that using stopwords-based passage priors (SW1 and SW2) to rank passages is novel to this study. We see that SW1 is the second most important feature for both INEX and AQUAINT. Another observation is that, as expected, the relative ordering of passages in this analysis, and the set of features that are among the top-10, are not identical to those presented above when using the passage features for document retrieval.

In addition, we perform ablation tests for PsgLTR. When using passages of 300 terms for INEX, the features whose removal resulted in statistically significant performance drop of MAiP are: ESA, MaxPDSim, AvgPDSim, SW1. The features whose removal resulted in statistically significant performance drop of MAP for AQUAINT are: Ent, SW1, ESA, SW2. The features are ordered in both cases in a descending order of the performance drop. Given that the retrieval tasks over INEX (passage retrieval) and AQUAINT (sentence retrieval) are different, it is not surprising that the feature lists are a bit different. Yet, ESA and SW1 are in both cases among the most important features, which was also the case above in the SVM-based analysis.

6 Discussion of empirical findings

The empirical analysis presented in Sect. 5 sheds light on the importance of different aspects of the proposed passage-based retrieval methods. These aspects were summarized in Table 1.

The superiority of JPDs to RRF provided support to the merits of using an LTR approach to integrate document and passage information with respect to fusing retrieval scores produced by document-based and passage-based document retrieval.

The superiority of RRF to SMPD, and the statistically indistinguishable performance of JPDs which uses a single passage and its variant that uses two passages (JPD-2; see Sect. 5.1.4), attest to the merits of using a single passage of the document to directly affect

its final retrieval score. Furthermore, selecting the document's passage that is the most highly ranked to this end is superior to selecting another passage as demonstrated by the performance of JPDs with various passages. (See Sect. 5.1.3.)

Another important finding was that using passage features to learn a document ranking function is of much merit with respect to using only passages' rank position information; e.g., FPD and JPDs outperform RRF and SMPD. Using passage features and document features together (JPDs) is superior to using them separately (FPD) for learning a document ranking function.

Given the above, it is not a surprise that JPDs was the best performing method among those proposed. It uses the features of the document's passage most highly ranked, together with the document features, to learn a document ranking function.

It is also important to note a relative merit of the proposed feature-based methods with respect to neural-network-based methods. The proposed methods were trained using a small number of queries (and relevance judgments) and yet outperformed highly effective baselines. As already noted above, it was shown that training even relatively simple neural networks for document ranking using the query sets we use here results in performance that is inferior to that of Okapi BM25 (Dehghani et al. 2017). Our best performing methods substantially outperform not only Okapi BM25, but also relevance model #3 (RM3) (Abdul-Jaleel et al. 2004); RM3 was shown to outperform some advanced neural network architectures for document retrieval via a system-to-system comparison (Lin 2018).

7 Conclusions and future work

Our focus in this work was on passage-based document retrieval: document ranking methods that utilize information induced from document passages. Previous work on passage-based document retrieval has focused on methods that integrate passage-query and document-query similarity values. Here, we addressed the challenge of utilizing richer sources of passage-based information for improving document retrieval effectiveness.

We presented a suite of learning-to-rank methods for document retrieval that use passage-based information. Most of the methods rely on ranking passages in response to the query using an effective approach, specifically, utilizing learning-to-rank. Some of the methods use information about the ranking of the passages of a document. Other methods use the passage-based features utilized for passage ranking and integrate them with document-based features so as to learn a document ranking function. We described connections between our methods and past unsupervised approaches for passage-based document retrieval as well as approaches for ranking clusters of similar documents.

To learn a passage-ranking method, we used previously proposed features along with features which were not used before for learning passage ranking functions. These features are query-independent passage-relevance priors adopted from work on using document relevance priors for Web search.

Empirical evaluation performed with a suite of datasets demonstrated the effectiveness of our methods. Our most effective method integrates document-based features with passage-based features of the document's most highly ranked passage. In addition, our best performing method was shown to outperform the use of different sets of document-based features. Further exploration provided support to the merits of using an effective passage

ranking method. We also showed that our passage-ranking method yields state-of-the-art passage retrieval performance.

For future work we intend to integrate in our methods additional passage-based features; e.g., those induced from inter-passage similarities (cf., Sheetrit and Kurland (2019)). We also plan to explore how our methods can be used for, and with, pseudo-feedback-query expansion. A case in point, we can apply query expansion at the passage-level, document-level, or both, so as to enrich the feature set used. Applying our methods with additional datasets (e.g., MS MARCO (Nguyen et al. 2016)) is also a future direction we intend to pursue.

Acknowledgements We thank the reviewers for their comments and Fiana Raiber for her help. This paper was partially supported by the German Research Foundation (DFG) via the German-Israeli Project Cooperation (DIP, Grant DA 1600/1-1) and by the Israel Science Foundation (Grant No. 1136/17).

References

Abdul-Jaleel, N., Allan, J., Croft, W. B., Diaz, F., Larkey, L., Li, X., et al. (2004). UMASS at TREC 2004: Novelty and hard. In *Proceedings of TREC*.

Arvola, P., Geva, S., Kamps, J., Schenkel, R., Trotman, A., & Vainio, J. (2011). Overview of the INEX 2010 ad hoc track. In *Comparative evaluation of focused retrieval* (pp. 1–32).

Bendersky, M., & Kurland, O. (2008). Re-ranking search results using document-passage graphs. In *Proceedings of SIGIR* (pp. 853–854).

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

Bendersky, M., Croft, W. B., & Diao, Y. (2011). Quality-biased ranking of web documents. In *Proceedings of WSDM* (pp. 95–104).

Buffoni, D., Usunier, N., & Gallinari, P. (2010). Lip6 at INEX: OWPC for ad hoc track. In *Focused retrieval and evaluation* (pp. 59–69).

Burges, C. J. (2010). *From ranknet to lambdarank to lambdamart: An overview*. Microsoft Research: Technical report.

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

Carmel, D., Shtok, A., & Kurland, O. (2013). Position-based contextualization for passage retrieval. In *Proceedings of CIKM* (pp. 1241–1244).

Chen, R., Spina, D., Croft, W. B., Sanderson, M., & Scholer, F. (2015). Harnessing semantics for answer sentence retrieval. In *Proceedings of ESAIR* (pp. 21–27).

Chen, R. C., Yulianti, E., Sanderson, M., & Cro, W. B. (2017). On the benefit of incorporating external features in a neural architecture for answer sentence selection. In *Proceedings of SIGIR* (pp. 1017–1020).

Cormack, G. V., Clarke, C. L., & Buettcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of SIGIR* (pp. 758–759).

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

Dehghani, M., Zamani, H., Severyn, A., Kamps, J., & Croft, W. B. (2017). Neural ranking models with weak supervision. In *Proceedings of SIGIR* (pp. 65–74).

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

Fan, Y., Guo, J., Lan, Y., Xu, J., Zhai, C., & Cheng, X. (2018). Modeling diverse relevance patterns in ad-hoc retrieval. In *Proceedings of SIGIR* (pp. 375–384).

Fernández, R. T., & Losada, D. E. (2012). Effective sentence retrieval based on query-independent evidence. *Information Processing and Management*, *48*(6), 1203–1229.

Fernández, R. T., Losada, D. E., & Azzopardi, L. A. (2011). Extending the language modeling framework for sentence retrieval to include local context. *Information Retrieval*, *14*(4), 355–389.

Ferragina, P., & Scaiella, U. (2012). Fast and accurate annotation of short texts with wikipedia pages. *IEEE Software*, *29*(1), 70–75.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*, 1189–1232.

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. *Proceedings of IJCAI*, *7*, 1606–1611.
Geva, S., Kamps, J., Lethonen, M., Schenkel, R., Thom, J. A., & Trotman, A. (2010). Overview of the inex 2009 ad hoc track. In *Focused retrieval and evaluation* (pp. 4–25).
Hearst, M. A., & Plaunt, C. (1993). Subtopic structuring for full-length document access. In *Proceedings of SIGIR* (pp. 59–68).
Jiang, J., & Zhai, C. (2004). Uiuc in hard 2004—passage retrieval using HMMS. In: TREC.
Joachims, T. (2006). Training linear SVMs in linear time. In *Proceedings of KDD* (pp. 217–226).
Kaszkiel, M., & Zobel, J. (1997). Passage retrieval revisited. In *Proceedings of SIGIR* (pp. 178–185).
Kaszkiel, M., & Zobel, J. (2001). Effective ranking with arbitrary passages. *Journal of the American Society for Information Science and Technology*, *52*(4), 344–364.
Keikha, M., Park, J. H., & Croft, W. B. (2014a). Evaluating answer passages using summarization measures. In *Proceedings of SIGIR* (pp. 963–966).
Keikha, M., Park, J. H., Croft, W. B., & Sanderson, M. (2014b). Retrieving passages and finding answers. In *Proceedings of ADCS* (p. 81).
Krikon, E., Kurland, O., & Bendersky, M. (2010). Utilizing inter-passage and inter-document similarities for reranking search results. *ACM Transactions on Information Systems*, *29*(1), 3:1–3:28.
Kurland, O., & Domshlak, C. (2008). A rank-aggregation approach to searching for optimal query-specific clusters. In *Proceedings of SIGIR* (pp. 547–554).
Kurland, O., & Krikon, E. (2011). The opposite of smoothing: A language model approach to ranking query-specific document clusters. *Journal of Artificial Intelligence Research*, *41*, 367–395.
Lang, H., Metzler, D., Wang, B., & Li, J. (2010). Improved latent concept expansion using hierarchical markov random fields. In *Proceedings of CIKM* (pp. 249–258).
Lin, J. (2018). The neural hype and comparisons against weak baselines. *SIGIR Forum*, *52*(2), 40–51.
Liu, T. Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, *3*(3), 225–331.
Liu, X., & Croft, WB. (2002). Passage retrieval based on language models. In *Proceedings of CIKM* (pp. 375–382).
Liu, X., & Croft, WB. (2004). Cluster-based retrieval using language models. In *Proceedings of SIGIR* (pp. 186–193).
Lv, Y., & Zhai, C. (2009). Positional language models for information retrieval. In *Proceedings of SIGIR* (pp. 299–306).
Lv, Y., & Zhai, C. (2010). Positional relevance model for pseudo-relevance feedback. In *Proceedings of SIGIR* (pp. 579–586).
Macdonald, C., Santos, R. L., & Ounis, I. (2012). On the usefulness of query features for learning to rank. In *Proceedings of CIKM* (pp. 2559–2562).
Metzler, D., & Croft, W. B. (2005). A markov random field model for term dependencies. In *Proceedings of SIGIR* (pp. 472–479).
Metzler, D., & Croft, W. B. (2007a). Latent concept expansion using markov random fields. In *Proceedings of SIGIR* (pp. 311–318).
Metzler, D., & Croft, W. B. (2007b). Linear feature-based models for information retrieval. *Information Retrieval*, *10*(3), 257–274.
Metzler, D., & Kanungo, T. (2008). Machine learned sentence selection strategies for query-biased summarization. In *Proceedings of SIGIR* (pp. 40–47).
Miao, J., Huang, J. X., & Ye, Z. (2012). Proximity-based rocchio's model for pseudo relevance. In *Proceedings of SIGIR* (pp. 535–544).
Mittendorf, E., & Schäuble, P. (1994). Document and passage retrieval based on hidden markov models. In *Proceedings of SIGIR* (pp. 318–327). New York: Springer.
Murdock, V., & Croft, W. B. (2005). A translation model for sentence retrieval. In *Proceedings of HLT/EMNLP* (pp. 684–691). Association for Computational Linguistics.
Murdock, V. G. (2006). Aspects of sentence retrieval. PhD thesis, University of Massachusetts Amherst.
Na, S., Kang, I., Lee, Y., & Lee, J. (2008). Completely-arbitrary passage retrieval in language modeling approach. In *Proceedings of AIRS* (pp. 22–33).
Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., et al. (2016). MS MARCO: A human generated machine reading comprehension dataset. *CoRR*. arXiv:1611.09268.
Ntoulas, A., Najork, M., Manasse, M., & Fetterly, D. (2006). Detecting spam web pages through content analysis. In *Proceedings of WWW* (pp. 83–92).
Raiber, F., & Kurland, O. (2013). Ranking document clusters using markov random fields. In *Proceedings of SIGIR* (pp. 333–342).

Raifer, N., Raiber, F., Tennenholtz, M., & Kurland, O. (2017). Information retrieval meets game theory: The ranking competition between documents? authors. In *Proceedings of SIGIR* (pp. 465–474).
Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). Okapi at trec-3. In *Proceedings of TREC* (Vol. 109, p. 109).
Salton, G., Allan, J., & Buckley, C. (1993). Approaches to passage retrieval in full text information systems. In *Proceedings of SIGIR* (pp. 49–58).
Sheetrit, E., & Kurland, O. (2019). Cluster-based focused retrieval. In *Proceedings of CIKM* (pp. 2305–2308).
Soboroff, I. (2004). Overview of the TREC 2004 novelty track. In *Proceedings of TREC*.
Soboroff, I., & Harman, D. (2003). Overview of the TREC 2003 novelty track. In *Proceedings of TREC* (pp. 38–53).
Tao, T., & Zhai, C. (2007). An exploration of proximity measures in information retrieval. In *Proceedings of SIGIR* (pp. 295–302).
Voorhees, E. M., & Harman, D. K. (2005). *TREC: Experiments and evaluation in information retrieval*. Cambridge: MIT Press.
Wan, X., Yang, J., & Xiao, J. (2008). Towards a unified approach to document similarity search using manifold-ranking of blocks. *Information Processing and Management*, *44*(3), 1032–1048.
Wang, M., & Si, L. (2008). Discriminative probabilistic models for passage based retrieval. In *Proceedings of SIGIR* (pp. 419–426).
Wilkinson, R. (1994). Effective retrieval of structured documents. In *Proceedings of SIGIR* (pp. 311–317).
Yang, L., Ai, Q., Spina, D., Chen, R. C., Pang, L., Croft, W. B., Guo, J., & Scholer, F. (2016). Beyond factoid QA: Effective methods for non-factoid answer sentence retrieval. In *Proceedings of ECIR* (pp. 115–128). Berlin: Springer.
Yulianti, E., Chen, R., Scholer, F., & Sanderson, M. (2016). Using semantic and context features for answer summary extraction. In *Proceedings of ADCS* (pp. 81–84).
Yulianti, E., Chen, R., Scholer, F., Croft, W. B., & Sanderson, M. (2018). Ranking documents by answer-passage quality. In *Proceedings of SIGIR* (pp. 335–344).
Zhai, C., & Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR* (pp. 334–342).
Zhao, J., & Yun, Y. (2009). A proximity language model for information retrieval. In *Proceedings of SIGIR* (pp. 291–298).