Sentence-Based Relevance Flow Analysis for High Accuracy Retrieval

Jung-Tae Lee
Department of Computer and Radio Communications Engineering, Korea University, 1, 5-ga, Anam-dong, Seongbuk-gu, Seoul, 136-713, South Korea. E-mail: jtlee@nlp.korea.ac.kr

Jangwon Seo
Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, MA 01003. E-mail: jangwon@cs.umass.edu

Jiwoon Jeon
Google Inc., Mountain View, CA 94043. E-mail: jjeon@google.com

Hae-Chang Rim*
Division of Computer and Communications Engineering, Korea University, 1, 5-ga, Anam-dong, Seongbuk-gu, Seoul, 136-713, South Korea. E-mail: rim@nlp.korea.ac.kr

Traditional ranking models for information retrieval lack the ability to make a clear distinction between relevant and nonrelevant documents at top ranks if both have similar bag-of-words representations with regard to a user query. We aim to go beyond the bag-of-words approach to document ranking in a new perspective, by representing each document as a sequence of sentences. We begin with an assumption that relevant documents are distinguishable from nonrelevant ones by sequential patterns of relevance degrees of sentences to a query. We introduce the notion of *relevance flow*, which refers to a stream of sentence-query relevance within a document. We then present a framework to learn a function for ranking documents effectively based on various features extracted from their relevance flows and leverage the output to enhance existing retrieval models. We validate the effectiveness of our approach by performing a number of retrieval experiments on three standard test collections, each comprising a different type of document: news articles, medical references, and blog posts. Experimental results demonstrate that the proposed approach can improve the retrieval performance at the top ranks significantly as compared with the state-of-the-art retrieval models regardless of document type.

Introduction

A term (i.e., word) has been the basic unit for document processing in many natural language processing and information retrieval (IR) tasks. Most conventional document retrieval models rely on simple term-statistics measured at the document level and/or at the collection level. These term statistics are precalculated and stored in a repository (e.g., index) for fast lookup. Search engines combine these term-statistics using various formulas to calculate the relevance scores of individual documents with regard to a given query.

The bag-of-words model has been most widely used to represent documents with respect to terms because of its simplicity and good performance demonstrated in various tasks. In this model, term orders and positions within a document are ignored, and each query term is treated independently. Although this simple model often has been successful for recall-oriented search applications, the overly simplified assumptions make it challenging to implement high accuracy retrieval systems (i.e., those capable of achieving high precision at top ranks). For example, the model has limitations in that nonrelevant documents containing many query terms by chance or in the wrong context can be ranked high because their bag-of-words representations would appear to be similar to those of relevant documents.

In this work, we shift the focus from term to sentence and propose a new sentence-oriented document representation that enables more complex analysis of document content and structure. Given a query, we first segment a document into sentences and calculate the relevance score of each sentence using conventional retrieval models. After normalizing the scores, the document can be viewed as a sequence of relevance scores with respect to the query, as a time-series.

Received May 28, 2010; revised March 30, 2010; accepted March 30, 2010

*Corresponding author.

© 2011 ASIS&T • Published online 27 May 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.21564

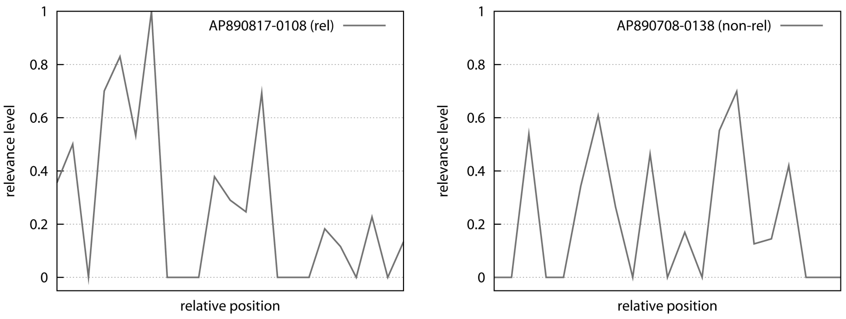


FIG. 1. Relevance flow graph of two top-ranked documents for TREC topic #102. The left and right respectively correspond to a relevant and a nonrelevant document ranked in the top two of the search result by a state-of-the-art retrieval model.

We can visualize the fluctuations of relevance scores depending on the positions of sentences. Figure 1 shows some sample graphs. We call them *relevance flows*.

Our basic assumption is that relevant documents have relevance flow patterns that are distinguishable from nonrelevant documents. For example, we observe that relevant documents often have a very high peak that implies a high density of query terms in a sentence; this suggests the proximity of query terms. Moreover, some types of documents, such as news articles, put important information or summaries at the beginning of their articles; the presence of early high peaks thus may be an informative source for inferring relevance for such types of documents. Our experimental results and analysis show that the proposed method can successfully integrate these human intuitions into the retrieval framework and improve the precision at top ranks. This type of document understanding is virtually impossible in traditional bag-of-words retrieval models.

To learn meaningful relevance flow patterns, we train Ranking Support Vector Machine (SVM) models with a collection of top-ranked documents retrieved by training queries. The labels of documents in the training set are determined by using relevance judgments of the training queries. Given a test query, documents are ranked according to the weighted sum of the scores of a baseline retrieval model and the Ranking SVM. We validate our approach on three different document collections: news, medical records, and blogs. We observe performance improvements in all the collections.

Related Work

Traditional term statistics-based approaches are often efficient but sacrifice the potential of structures embedded in documents. To overcome this drawback, there have been numerous efforts made to exploit document structures. For example, there are a number of studies that consider positions of query terms in documents (i.e., term proximity). Keen (1992)'s work is among the earliest that suggested the idea of term proximity. He proposed several simple non-Boolean ranking methods based on term proximity and compared them with traditional Boolean systems. Some work tried to integrate proximity features into bag-of-words ranking models. Rasolofo and Savoy (2003) added proximity information to the Okapi probabilistic model and found improved performance specifically among the top scored documents. Buttcher, Clarke, and Lushman (2006) also incorporated proximity into the Okapi BM25 model and observed positive results. Metzler and Croft (2005) introduced the use of the Markov random fields (MRF) for modeling term dependencies in the language-modeling framework. They introduced three variants of the MRF, where each captured different dependencies between query terms, and demonstrated that modeling the dependencies can improve retrieval effectiveness. All of these studies demonstrated performance improvements in various document retrieval tasks and proved that proximity can become an effective feature. However, their techniques are not flexibly applicable to other models or tasks in that they are designed to work in specific retrieval frameworks.

Some recent work focused more on features to measure term proximity. Tao and Zhai (2007) explored several proximity measures and designed heuristic constraints for incorporating proximity measures into an existing retrieval model. In their experiments, one of the proposed proximity measures was shown helpful in improving the retrieval performance of the KL-divergence language model and the Okapi BM25 model. Cummins and O'Riordan (2009) used a learning approach to combine various proximity measures to obtain an effective proximity-based retrieval function. Their approach outperformed both the traditional Okapi BM25 model and the approach of Tao and Zhai (2007). Zhao and Yun (2009) also developed several forms of proximity measures for scoring the proximity of individual query terms. They combined the proximity information with language models and showed improved retrieval performance. Similar to our work, these techniques tend to be more flexible than the previous ones, because they are more based on features rather than specific retrieval models or frameworks.

Some techniques considered distributional patterns of query terms in documents rather than addressing proximity explicitly. These techniques can be more efficient than proximity-based methods in that they use discriminative features rather than addressing dependencies between all terms. Earlier studies on passage retrieval (Callan, 1994; Kaszkiel & Zobel, 1997) combined local term statistics observed from passages of a document. They showed that the use of fixed-length passages could bring improvements in retrieval effectiveness particularly for long documents. Troy and Zhang (2007) focused on the position of the earliest occurrence of the query terms to enhance ranking techniques and showed positive results compared to the Okapi BM25 and the language modeling approaches. Xue and Zhou (2009) used distributional pattern-based features, which include the compactness of the appearance of the word and the position of the first appearance of the word, for text categorization. However, although these studies are similar to our work in that they use distributional patterns, they focus on a few aspects rather than various features and relations between such aspects. The work of Seo and Jeon (2009) is also similar to our work in that they consider sentence-level distributional patterns. However, the work is yet preliminary and needs further verification, because their experiments are conducted on a small test collection with a few features. Recently, Lv and Zhai (2009) suggested the positional language model, which estimates distributions of terms according to their positions, and showed its effectiveness over basic language models. They also compared their model with the approach of passage retrieval and Tao and Zhai (2007); however, the model did not outperform previous approaches on some test collections. While their work focuses on generative representations, we try to directly learn document relevance using various discriminative features obtained from relations between queries and document structures.

In addition, to address multiple features in a flexible framework, effective feature combination techniques are necessary. Combination heuristics introduced by Fox and Shaw (1994) have been widely used in many IR tasks. In recent years, a large number of learning-to-rank techniques have been suggested to pursue scalability in terms of the size of training data and the number of features (Burges et al., 2005; Joachims, 2002b).

Proposed Method

Throughout this section, we assume that the following have been given: a query q and a set D_q of documents most highly ranked by some initial retrieval algorithm in response to q. The aim of this study is to (re-)rank D_q so that more relevant documents are ranked higher than less relevant or nonrelevant ones in the result displayed to the user. We propose to do this by leveraging the information about the flow of relevance within each document $d_i \in D_q$ represented as multiple features. In particular, we adopt machine learning techniques to train a function that ranks D_q based on the features.

Relevance Flow Extraction

Formally, a document $d_i \in D_q$ can be represented as a sequence of sentences as:

$$(s_1, s_2, s_3, \cdots, s_n)$$

where s_j corresponds to the sentence at absolute position j and n represents the number of sentences in d_i respectively.

The *relevance flow* of d_i with respect to q is extracted by computing the relevance of individual sentences with some scoring function, as:

$$(l_1, l_2, l_3, \cdots, l_n)$$

where l_j represents the relevance score of the sentence s_j normalized to the range 0 to 1 using the minimum and maximum sentence scores across all documents in D_q. We refer to this score as the *relevance level* of s_j. We also define a sentence with relevance level higher than a pre-defined value α as a *peak*, where $0 \leq \alpha < 1$.

This form of document representation reflects the oscillation of relevance level within a document with regard to a query. Despite its simplicity, relevance flows may provide some useful evidence for predicting document relevance. Figure 1 visually illustrates the relevance flows of a relevant document and a nonrelevant document within top part of the retrieved list for a same query. We show them by plotting the relevance levels of individual sentences versus their relative positions within each document.

From the illustrations, some clear differences between the two can be easily noticeable. For example, the relevance flow on the left has high peaks in the very beginning, while the one on the right has many moderate peaks across the document. The figure suggests that relevance flow patterns may be useful for discriminating relevant documents from nonrelevant ones if such meaningful information could be learned from a number of query-document pairs in advance. We will discuss our learning framework in detail in the following subsections.

Sentence Scoring Function

Given a definition of relevance flow, a scoring function for computing the relevance score of individual sentences must be chosen. However, there are two issues: variability in sentence length and the scale of term frequency in sentences. Because sentences have different lengths, normalization may be problematic. Moreover, because sentences are shorter than documents, same words would not occur repetitively in sentences; thus, considering raw frequencies may be useless.

We therefore use a simplified variant of the Okapi BM25 model (Jones, Walker, & Robertson, 2000) as the scoring function, which has two free parameters for controlling the issues above. The relevance score of a sentence s given q is calculated as:

$$\sum_{i \in q} \frac{(k_1 + 1) \cdot tf_i}{tf_i + k_1 \cdot \left(1 - b + b \cdot \frac{|s|}{avsl}\right)} \cdot \log \frac{N}{sf_i + 1}$$

where tf_i is the frequency of query term i in s; $|s|$ is the length of s in words; *avsl* is the average length of a sentence in the collection; N is the total number of sentences in the collection; and sf_i is the number of sentences containing i. The constant b regulates the impact of sentence length normalization; $b = 0$ corresponds to no normalization, and $b = 1$ is full normalization. The constant k_1 calibrates the sentence term frequency scaling; $k_1 = 0$ corresponds to a binary model where only term presence/absence would matter, while setting a large value for k_1 corresponds to using raw frequency.

Features

We now define a variety of features that we hypothesize would be useful for representing relevance flows of individual documents. The features intuitively embody the two main aspects of the relevance flow of a document: the relevancy and the locality information. Note that we normalize the position of each relevance level in the range of 0 and 1 before constructing a feature vector. For example, the relative positions of the first and the last sentences in a document would be 0 and 1, respectively.

- SumRel: The sum of all relevance levels within the document.
- AveRel: The average relevance level within the document. This set comprises the arithmetic mean and the harmonic mean of all relevance levels.
- AvePeakRel: The average relevance level of peaks. This set also comprises both the arithmetic and the harmonic mean.
- PeakRatio: The ratio of peaks to the document length (in sentences).
- HighPeakRel: The relevance level of the highest peak. A highest peak would correspond to a sentence containing the highest density of query terms.
- VarRel: The amount of variation within the relevance levels. This set comprises the variance, the standard deviation, and the variance-to-mean ratio of the relevance levels. Higher values would correspond to more undulations (i.e., ups and downs) in the relevance levels.
- VarPeakRel: The amount of variation within the relevance levels of peaks. This set comprises the variance, the standard deviation, the range, and the variance-to-mean ratio.
- FirstPeakPos: The relative position of the first peak. A small value would indicate that the document has an early peak; this would imply that the document contains a title or an introductory sentence that is relevant to the query.
- LastPeakPos: The relative position of the last peak.
- AvePeakPos: The average relative position of peaks. This roughly indicates where peaks usually appear in the document.
- HighPeakPos: The relative position of the highest peak.
- VarPeakPos: The variance of the relative positions of peaks. A lower value would indicate that peaks occur close to each other, and a higher value would indicate that they occur far away from each other.
- PeakSpan: The distance between the first peak and the last peak in the document. This indicates the ratio of sentences covered by the two peaks to the document length (in sentences).
- PeakNeighbor: The average relevance level of sentences neighboring the peaks. A higher value would correspond to a higher cohesiveness of query related sentences in the document.
- ConsecPeak: Recall that we call a sentence with relevance level higher than α as a peak. If two or more continuously positioned sentences have relevance levels higher than α, we call them *consecutive peaks*. This set of features comprises the ratio of all consecutive peaks to the document length as well as the ratio of the maximum consecutive peaks to the document length. These features also roughly indicate the cohesiveness of relevant sentences in the document.

Learning Mechanism

In this article, we adopt the Ranking SVMs (RSVM; Herbrich, Graepel, & Obermayer, 2000; Joachims, 2002b) for learning a ranking function based on the features suggested above. The main reason why we use them is that they are based on pairwise learning, which means that they can be generalized to any graded relevance scale of documents (including the binary scale). Note that our goal is not to evaluate a range of existing learning-to-rank approaches but to rather demonstrate how such method can be applied successfully to the task at hand.

Here we will give a very brief description of how RSVM works. Assume that there is a training set S that comprises (q_i, r_i) tuples, where q_i corresponds to the ith query, and r_i corresponds to the pairwise preference information of q_i. For example, if document d_1 should be ranked higher than document d_2 for q_i, then $\{d_1 > d_2\} \in r_i$. Given the set S, we want to learn a ranking function f such that:

$$d_i > d_j \Leftrightarrow f(\Phi_i) > f(\Phi_j)$$

where Φ_i and Φ_j are feature vectors of documents d_i and d_j respectively. Assume that f is a linear function:

$$f(\Phi) = \mathbb{W} \cdot \Phi$$

where $\mathbb{W}$ is a weight vector. If we combine the two equations above, we get:

$$d_i > d_j \Leftrightarrow \mathbb{W} \cdot (\Phi_i - \Phi_j) > 0$$

Note that the pairwise preference information $d_i > d_j$ is now expressed by a new vector $\Phi_i - \Phi_j$. Accordingly, we can create a new training set S' from S that contains new vectors $\Phi_i - \Phi_j$ and their labels $+1$ or -1 as $d_i > d_j$ or $d_i < d_j$, respectively. We can then construct a binary SVM classifier that classifies a vector $\Phi_i - \Phi_j$ as $+1$ or -1. See the work of Joachims (2002a, 2002b, 2006) for details on finding solutions for the weight vector $\mathbb{W}$. Once $\mathbb{W}$ is found, we can obtain a new relevance score for the document $d_i \in D_q$ by $\mathbb{W} \cdot \Phi_i$.

To construct the ground truth for training RSVM, we use a set of top-ranked documents retrieved by some ranking model in response to a set of training queries. We use available relevance judgments to create the pairwise preference information so that documents judged to be more relevant are put above those judged to be less relevant or nonrelevant. For example, if there are two documents, d_1 and d_2, retrieved with regard to a training query and their relevance judgments are available (e.g., d_1 is relevant whereas d_2 is not), we can automatically create a pairwise preference tuple of the two documents (i.e. $\{d_1 > d_2\}$), because users would prefer to see relevant documents placed at higher rankings than nonrelevant ones in the retrieved result.

TABLE 1. Test collection statistics.

Test collection	#Docs	#Wrds/Doc	#Snts/Doc	#Wrds/Query
AP88-90	242,918	463	12	5
OHSUMED	348,566	127	13	7
BLOGS06	3,215,171	1,290	23	2

Note. Stop word removal and stemming are performed on documents.

Final Ranking

To verify the effectiveness of the newly predicted relevance score of each document $d_i \in D_q$, we linearly combine them with the initial scores given by either of the following two representative state-of-the-art ranking models, the Okapi BM25 model (Jones et al., 2000) and the query likelihood language model (Croft, Metzler, & Strohman, 2009) with Dirichlet smoothing (Zhai & Lafferty, 2004), as follows:

$$Score(d_i, q) = (1 - \lambda) \cdot Score_{init} + \lambda \cdot Score_{new}$$

where $Score_{init}$ is the initial score of d_i with regard to q given by either of the two baseline models; $Score_{new}$ is the new relevance score of d_i predicted by the RSVM model; and λ is a constant for regulating the impact of $Score_{new}$ in ranking, where $0 \leq \lambda \leq 1$. Before the linear combination, we normalize both the $Score_{init}$ and $Score_{new}$ scores in the range of 0 to 1 using the maximum/minimum $Score_{init}$ and $Score_{new}$ scores in D_q, respectively.

Empirical Evaluation

Setup

Experiments are conducted on the following three standard test collections: AP88-90, OHSUMED, and BLOGS06. Table 1 shows the statistics of individual test collections.

AP88-90 (Harman, 1994) includes newswire articles from the Associated Press (1988–1990) in Text REtrieval Conference (TREC) disk 1-3, with average 463 words per document. The *title* field of the three TREC topic sets (51–100, 101–150, 151–200) is used as query; it is longer than typical keyword queries for web search.

OHSUMED (Hersh, Buckley, Leone, & Hickam, 1994) contains relatively short abstracts of references from medical journals in the MEDLINE database. The average length of sentences is similar to AP88-90. These data have been used in many retrieval experiments, including the TREC-9 Filtering Track and the Special Interest Group on Information Retrieval (SIGIR) Workshops on Learning to Rank (LR2IR). Note that we have used the *information request* field of the OHSUMED topics 1-106 query; it is more verbose than queries in AP88-90.

BLOGS06 (Ounis, de Rijke, Macdonald, Mishne, & Soboroff, 2006) refers to a large-scale collection of blog posts used in the TREC Blog Track 2006–2008. Only the *title* field of the three TREC topic sets (851–900, 901–950, 1001–1050) is used as query. In our experiment, we regard a blog post as relevant as long as it is assessed to be *topically* relevant (i.e., scale 1 or above) in the relevance judgment set. Compared with the other two collections, BLOGS06 contains extremely long documents (with long sentences) and short keyword queries with an average of two words.

In each experiment, we first use a baseline bag-of-words model (either the Okapi BM25 model or the query likelihood language model) with its parameters set to their best values to retrieve the top 15 documents for each query and then use the proposed method to re-rank them. The accuracy of the top-ranked documents for both the baseline run and the proposed run are compared with the normalized discounted cumulative gain (NDCG) measure (Järvelin & Kekalainen, 2002), which gives a high evaluation score to a ranked list where relevant documents are ranked higher than nonrelevant documents. The NDCG at a particular rank n is calculated as follows:

$$NDCG@n = Z_n \cdot \sum_{i=1}^{n} (2^{rel_i} - 1)/\log(1 + i)$$

where Z_n is the normalization constant so that the perfect ranking is evaluated as 1, and rel_i is the relevance of the document retrieved at rank i.

The reason why we consider only a small number of top documents for re-ranking is two-fold. First, we intuitively assume that in many cases search users would choose to see only the first page and sometimes the second page of search results where each result page usually contains 10 documents. Second, the proposed method involves automatic sentence segmentations and feature extractions that require additional computation; increasing the number of top documents to be re-ranked would linearly increase the amount of computation at the retrieval stage, which would make the method impractical.

When there is not a large amount of query data for training, K-fold cross validation is done in practice by partitioning the data into K subsets and using $K - 1$ subsets for training and the remaining subset for testing. This is repeated K times with each subset used once for testing, and the K results are averaged. In this article, three-fold cross validation is performed on AP88-90 and BLOGS06, regarding each query set as a fold. Five-fold cross validation is performed on OHSUMED because of less number of topics compared with the other two collections.

The Lemur Toolkit (Ogilvie & Callan, 2001) is used for indexing and retrieval. All collections are stemmed using the Porter stemmer (Porter, 1980) and "stopped" using a list of 418 stop words in Lemur. The segmentation of document text into sentence units is done using the publicly available sentence segmentation tool developed by the Cognitive Computation Group at UIUC (Munoz & Nagarajan, n.d.). (For BLOGS06 collection, we automatically remove the HTML tags from documents before the segmentation.)

SVM-rank (Joachims, 2006) is used to learn RSVM models based on the relevance flow features of documents. For all experiments in this article, we use the linear kernel function. We have chosen the SVM regularization parameter C in the set {0.001, 0.01, 0.1, 1, 10, 100, 1000}. All performance

TABLE 2. Best performance of the proposed method.

Collection	Method	NDCG@5	NDCG@10
AP88-90	BM25 ($k_1 = 2$, $b = 0$)	0.3292	0.3236
	BM25+RF ($k_1 = 0$, $b = 0$, $\alpha = 0.1$, $\lambda = 0.5$)	**0.3450** (+5%)†	**0.3333** (+3%)†
	LM ($\mu = 5000$)	0.3235	0.3185
	LM+RF ($k_1 = 1.2$, $b = 0.5$, $\alpha = 0.05$, $\lambda = 0.3$)	**0.3353** (+4%)	**0.3307** (+4%)†
OHSUMED	BM25 ($k_1 = 1.2$, $b = 0.7$)	0.4183	0.4060
	BM25+RF ($k_1 = 1.2$, $b = 1$, $\alpha = 0.9$, $\lambda = 0.3$)	**0.4362** (+4%)	**0.4160** (+2%)†
	LM ($\mu = 300$)	0.4121	0.3913
	LM+RF ($k_1 = 1.2$, $b = 1$, $\alpha = 0.75$, $\lambda = 0.2$)	**0.4331** (+5%)‡	**0.4041** (+3%)‡
BLOGS06	BM25 ($k_1 = 1.2$, $b = 0.1$)	0.6047	0.6039
	BM25+RF ($k_1 = 1.2$, $b = 1$, $\alpha = 0.95$, $\lambda = 1$)	**0.6837** (+13%)‡	**0.6603** (+9%)‡
	LM ($\mu = 5000$)	0.5807	0.5905
	LM+RF ($k_1 = 1.2$, $b = 1$, $\alpha = 0.7$, $\lambda = 1$)	**0.6599** (+14%)‡	**0.6431** (+9%)‡

Note. RF = relevance flow score.
Highlighted figures correspond to the scores of proposed method.
† and ‡ indicate the improvement of the proposed method over the baseline is significant at $p < 0.1$ and $p < 0.05$ level, respectively.

figures in this article are derived by setting C to 0.1, which had demonstrated the best performance in our preliminary experiments. The number of top documents retrieved for each training query (for generating the training sets for RSVMs) is also set to 15 for all experiments.

Results

Best performance. In Table 2, we report the best retrieval performance of the proposed method with the two baseline bag-of-words models. Note that for each test collection, we have customized the baseline bag-of-words models in advance to maximize their NDCG@10 values by "tuning" the values for the parameters k_1 and b in the Okapi BM25 model and the Dirichlet smoothing parameter μ in the query likelihood language model. The parameter k_3 in the Okapi model is always set to seven as suggested by Robertson and Walker (1999). The best performance of the proposed method is found by varying the following four parameters: the sentence scoring function parameters k_1 and b, the peak level α, and the linear interpolation parameter λ; the optimal combinations of these four parameters for the proposed method runs are also shown in Table 2. We have performed significance tests on the improvements of the proposed method over the baselines using Student's t test, which is one of the most commonly used tests in the evaluation of retrieval models. Such tests enables us to reject a null hypothesis (there is no difference between the proposed and baseline methods) in favor of an alternative hypothesis (the proposed method is better than the baseline; Croft et al., 2009).

The proposed method outperforms both of the baseline bag-of-words models at the significance level of $p < 0.05$ or $p < 0.1$ in most runs on all three test collections, especially in terms of NDCG@10. Note that the performance of the proposed method is insensitive to either the choice of the initial retrieval model or the type of the document collection. The results show that the relevance flow information of documents with regard to a query is potentially effective in inferring document relevance and in improving the quality of top search results.

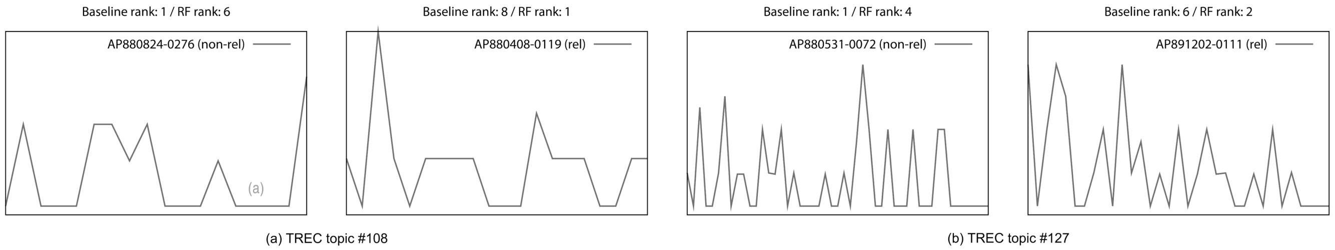
TABLE 3. Feature ablation results.

Feature	AP88-90	OHSUMED	BLOGS06
SumRel	**−3.57%**	−0.60%	+0.23%
AveRel	−0.45%	−0.67%	− 0.77%
AvePeakRel	+0.57%	**−4.83%**	**−1.56%**
PeakRatio	− 0.09%	−0.87%	−0.79%
HighPeakRel	**−5.13%**	**−1.32%**	**−1.11%**
VarRel	+0.15%	−0.12%	**−1.06%**
VarPeakRel	**−2.52%**	−0.05%	−0.80%
FirstPeakPos	+0.39%	−0.31%	**−1.09%**
LastPeakPos	+0.12%	−0.10%	−0.92%
AvePeakPos	−0.63%	−0.14%	−0.77%
HighPeakPos	**−2.31%**	−0.72%	**−1.79%**
VarPeakPos	**−1.05%**	−0.12%	−0.70%
PeakSpan	−0.66%	−0.02%	**−1.33%**
PeakNeighbor	−0.63%	+0.00%	−0.67%
ConsecPeak	−0.15%	−0.12%	−0.70%

Note. Each figure represents percentage change in NDCG@10 score by removing each feature. Highlighted figure corresponds to more than one percent loss in accuracy.

Individual feature contribution. We now turn to the important question: How do the different types of relevance flow features contribute to the overall retrieval performance? We study this problem by conducting a series of feature ablation experiments (i.e., removing each feature one at a time) on all three collections using the BM25+RF framework with all four parameters set to optimal values. The accuracy loss obtained by the removal of a feature roughly reflects the contribution of the feature. The results of feature ablation studies are shown in Table 3.

We were not surprised to find that many features show different degrees of contribution on individual collections. Nevertheless, the most dominant feature that contributes the



(a) TREC topic #108

(b) TREC topic #127

FIG. 2. Illustrations of real examples. The horizontal and vertical axes correspond to the relevant position and the relevance level, respectively. "Baseline rank" and "RF rank" correspond to the rank of each document in the retrieved result of the BM25 model and the RSVM model, respectively.

most on all three collections is the HighPeakRel feature, which we found has positive correlation to document relevance. This result is also consistent with the findings of earlier studies on passage retrieval; a document with a short block of highly relevant text is more likely to be relevant than one that contains a reasonable number of query terms across its length.

The interesting observation we have made is that the HighPeakPos feature also showed affirmative contributions. This suggests that not only is the presence of highly relevant piece of text important, but its position within the document is also valuable for inferring document relevance. We have found that the HighPeakPos feature has negative correlation to document relevance. This result is intuitively convincing, because many writers tend to place key topics at the front as titles or introductory sentences/paragraphs; a document would be more likely to be relevant if the relevant text is placed in the beginning of the document. This result also confirms similar findings in studies on incorporating chronological term positions in retrieval models (Troy & Zhang, 2007).

Other features related to peak positions, such as FirstPeakPos and LastPeakPos, also show supplementary contributions. They showed higher contributions to performance on BLOGS06 but insignificant effect on AP88-90 and especially OHSUMED. This result suggests that they tend to contribute more on collections with long documents.

We have also found that features that show not only the central tendencies of relevance levels within a document (e.g., AveRel and AvePeakRel) but also the dispersion in relevance levels (e.g., VarRel and VarPeakRel) are generally helpful. This suggests that documents having more indented relevance flows with high peaks in regard to the query are more likely to be relevant than ones showing less fluctuation in relevance levels.

Features that reflect the cohesiveness of peaks, such as VarPeakPos and PeakSpan, have been observed to be helpful for AP88-90 and BLOGS06 collections but not for OHSUMED. This is a reasonable result, because extremely short documents contain very few sentences; there would be less chance for relevant sentences to occur cohesively in such documents. The SumRel feature, on the other hand, shows negative contribution on BLOGS06. We believe this is because of the fact that the feature values for SumRel are not normalized to the document length, thus bringing negative effect to the performance on long documents.

Figure 2 demonstrates the effects of some relevance flow features by comparing the ranks of some relevant and nonrelevant documents in the retrieved lists of a baseline model (Okapi BM25) against the RF model (before the linear combination). In Figure 2a, both documents have considerable number of query terms across the document and also considerably high peaks. However, the left one (nonrelevant) has a high peak in the very end of the document whereas the right one (relevant) has its high peak in the beginning. The baseline model outputs an incorrect ranking by placing the nonrelevant one at the first rank, but the RF model successfully ranks them so that the relevant document is ranked first. Figure 2b is obviously more challenging because both documents have considerable number of high peaks. However, there is a difference in high peak positions; the right one (relevant) has relatively higher and wider peaks at the beginning compared with the left one (nonrelevant). Note that the baseline model misplaces the nonrelevant one at the first rank, but the RF model correctly orders them by successfully detecting the difference.

Techniques for sentence scoring function. The scoring function we have used for computing the relevance level of individual sentences has two free parameters, k_1 and b, for controlling the effect of sentence length normalization and term frequency scaling, respectively. We now investigate the sensitivity of the two parameters.

We first look into the sentence length normalization parameter b, by fixing it to certain values and observing their best performances achieved by varying the remaining three parameters. The following three settings have been tested: $b = 0$ (no normalization), $b = 0.5$, and $b = 1$ (full normalization). Figure 3 shows some representative results on three collections using the BM25+RF framework. All three settings outperform the baseline on all levels of NDCGs on all collections. However, the full normalization ($b = 1$) is shown to be more stable and accurate, especially on OHSUMED and BLOGS06. This suggests that sentence length normalization is important for inferring document relevance using sentences of varying length.

We now examine the sensitivity of parameter k_1 in the scoring function in a similar way. For simplicity, we have tested

DOI: 10.1002/asi

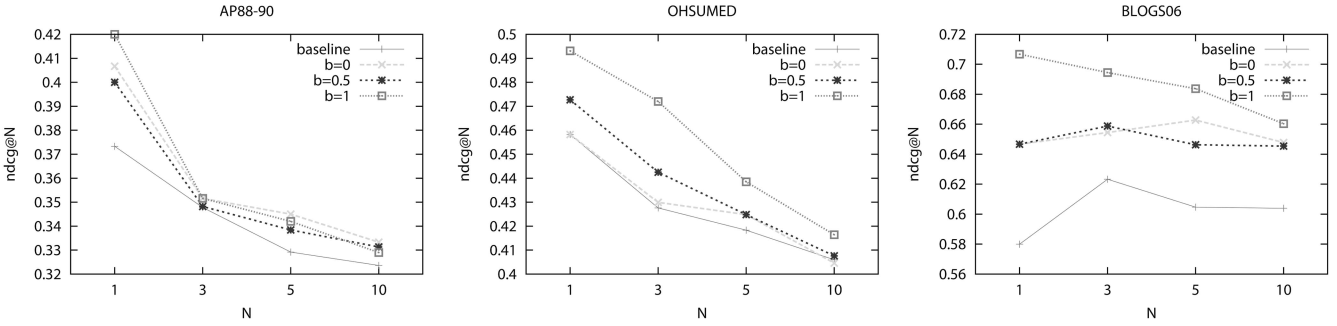


FIG. 3. Sensitivity to parameter b for length normalization in sentence scoring function.

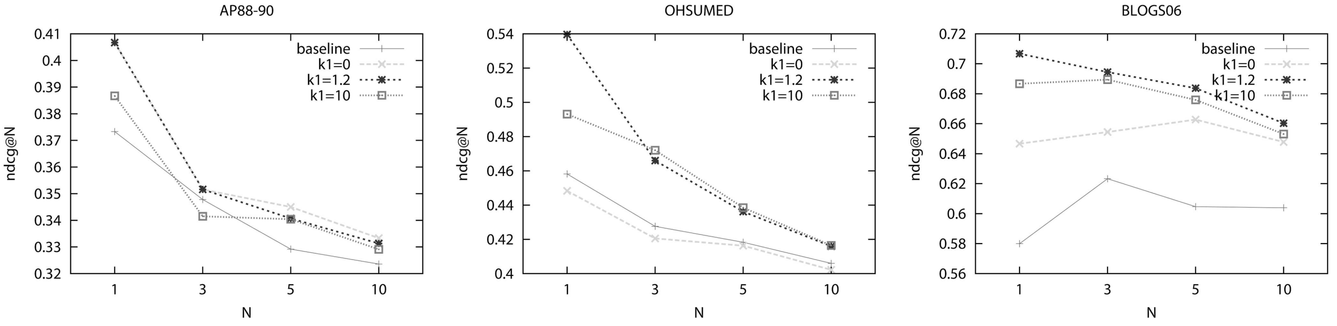


FIG. 4. Sensitivity to parameter k1 for term frequency scaling in sentence scoring function.

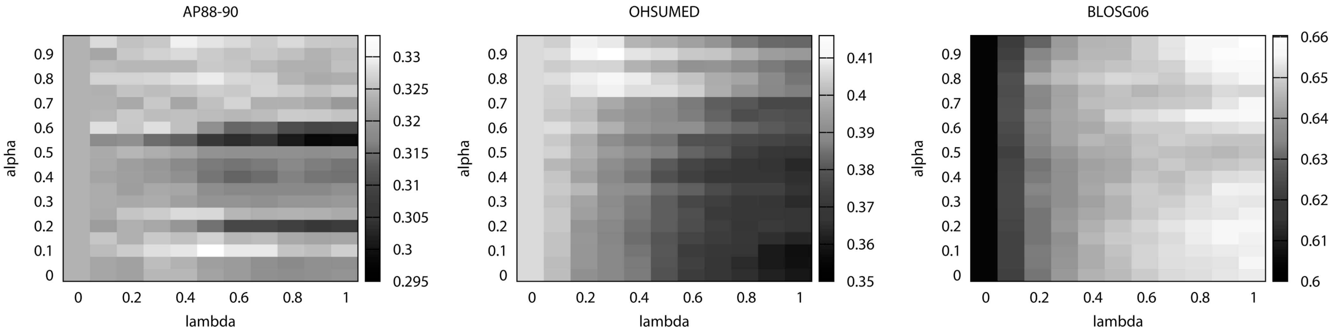


FIG. 5. Sensitivity to peak level parameter α and linear combination parameter λ. Lighter color represents better NDCG@10 performance.

the following three settings: $k_1 = 0$ (a binary model), $k_1 = 1.2$ (similar effect to log term frequency), and $k_1 = 10$ (a model considering raw term frequency). Figure 4 shows the results. The result is not much different from the usual document-unit retrieval setting. The performance is always stable when $k_1 = 1.2$, which causes the effect of tf_i to be very non-linear (i.e., high frequency would not have much impact). Even though the binary model ($k_1 = 0$) shows slightly better performance on AP88-90, the $k_1 = 1.2$ setting shows substantial improvement compared to the baseline.

Sensitivity to parameters α and λ. Last, we look into the sensitivity of the retrieval performance to the peak level parameter α and the linear combination weight λ in the proposed method. The heat maps (Weinstein, 2008) in Figure 5 show the influence of α and λ on the performance of the BM25+RF framework. Higher NDCG@10 values are represented by lighter squares and lower values by darker squares. We have observed that the sensitivity of α tends to correlate with the average length of documents; OHSUMED prefers high α values, but the performances on AP88-90 and BLOGS06 seem less sensitive to α. Based on this observation, we suspect that determination of peaks in a document should be performed more strictly (i.e., set α with high values) when dealing with relatively shorter documents. The parameter λ, which determines the impact of relevance flow analysis, shows a different tendency; better performance is shown on AP88-90 and OHSUMED with lower λ values but on BLOGS06 with high λ values. This suggests that the result of the relevance flow analysis is more stable with long documents. We find that the QL+RF framework has similar tendencies.

Conclusions

With the growing size of web document collections, high precision at the top of the search result has become an important issue for search engine users. However, traditional retrieval models have limitations in that a document is considered to be an unordered collection of words. This simple assumption has led many traditional models to be successful for recall-oriented search applications but virtually makes it difficult to implement high accuracy retrieval systems that require more different features of documents.

This article has presented a new approach to represent a document, or more precisely its structure, with regard to a query. The approach is to break down a given document into sentences and calculate their individual relevance toward a given query. Then, the document can be represented as the fluctuation of relevance with regard to the query; we have referred to this representation method as the relevance flow throughout the article. Our assumption is that query-relevant documents would have distinguishable relevance flow patterns from nonrelevant ones. The key insight of this article is that such sentence-level evidence can provide useful information for inferring document relevance against the query. Based on a novel set of features for characterizing the relevance flows of individual documents, we explored a machine learning framework to learn a preference function based on meaningful relevance flow patterns of relevant and nonrelevant documents. Experimental results on three different test collections show that the proposed method is capable of improving the accuracy of the top-ranked result significantly for various types of document collections compared with the state-of-the-art bag-of-words retrieval models used in practice, including the Okapi BM25 model and the language model with "tuned" parameters.

Directions of possible future work include exploring a wider range of text units such as paragraphs or fixed-length passages for analyzing relevance flows, exploring different training methods with use of large-scale click-through data, investigating automated ways for parameter optimization, and applying other suitable score combination methods.

Acknowledgment

This research was supported by the Second Brain Korea 21 Project. Any opinions, findings, and conclusions or recommendations expressed above are those of the authors and do not necessarily reflect the views of the sponsors.

References

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. Proceedings of the 22nd International Conference on Machine Learning (ICML '05) (pp. 89–96). New York: ACM Press.

Buttcher, S., Clarke, C.L.A., & Lushman, B. (2006). Term proximity scoring for ad-hoc retrieval on very large text collections. Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '06) (pp. 621–622). New York: ACM Press.

Callan, J.P. (1994). Passage-level evidence in document retrieval. Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94) (pp. 302–310). New York: Springer-Verlag.

Croft, W.B., Metzler, D., & Strohman, T. (2009). Search engines: Information retrieval in practice. Boston, MA: Addison Wesley.

Cummins, R., & O'Riordan, C. (2009). Learning in a pairwise term-term proximity framework for information retrieval. Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09) (pp. 251–258). New York: ACM Press.

Fox, E.A., & Shaw, J.A. (1994). Combination of multiple searches. Proceedings of the Second Text REtrieval Conference (TREC-2) (pp. 243–252). Alexandria, VA: NTIS.

Harman, D. (1994). Overview of the third text retrieval conference (TREC-3). Proceedings of the Third Text REtrieval Conference (TREC-3) (pp. 1–19). Alexandria, VA: NTIS.

Herbrich, R., Graepel, T., & Obermayer, K. (2000). Large margin rank boundaries for ordinal regression. In A.J. Smola, P.L. Bartlett, B. Schölkopf, & D. Schuurmans (Eds.), Advances in large margin classifiers (pp. 115–132). Cambridge, MA: MIT Press.

Hersh, W., Buckley, C., Leone, T.J., & Hickam, D. (1994). Ohsumed: An interactive retrieval evaluation and new large test collection for research. Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94) (pp. 192–201). New York: Springer-Verlag.

Järvelin, K., & Kekalainen, J. (2002). Cumulated gain-based evaluation of IR techniques. Transactions on Information Systems, 20(4), 422–446. doi:10.1145/582415.582418

Joachims, T. (2002a). Learning to classify text using support vector machines: Methods, theory and algorithms. Norwell, MA: Kluwer.

Joachims, T. (2002b). Optimizing search engines using clickthrough data. Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02) (pp. 133–142). New York: ACM Press.

Joachims, T. (2006). Training linear SVMS in linear time. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06) (pp. 217–226). New York: ACM Press.

Jones, K.S., Walker, S., & Robertson, S.E. (2000). A probabilistic model of information retrieval: Development and comparative experiments. Information Processing and Management, 36(6), 779–808. doi:10.1016/S0306-4573(00)00015-7

Kaszkiel, M., & Zobel, J. (1997). Passage retrieval revisited. Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07) (pp. 178–185). New York: ACM Press.

Keen, E.M. (1992). Term position ranking: Some new test results. Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '92) (pp. 66–76). New York: ACM Press.

Lv, Y., & Zhai, C. (2009). Positional language models for information retrieval. Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09) (pp. 299–306). New York: ACM Press.

Metzler, D., & Croft, W.B. (2005). A Markov random field model for term dependencies. Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05) (pp. 472–479). New York: ACM Press.

Munoz, M., & Nagarajan, R. (n.d.). Sentence segmentation tool. Retrieved from http://l2r.cs.uiuc.edu/∼cogcomp/tools.php

Ogilvie, P., & Callan, J. (2001). Experiments using the lemur toolkit. Paper presented at the 19th Text REtrieval Conference (TREC 2010), Gaithersburg, MD.

Ounis, I., de Rijke, M., Macdonald, C., Mishne, G., & Soboroff, I. (2006). Overview of the TREC 2006 blog track. Paper presented at the 15th Text REtrieval Conference (TREC 2006), Gaithersburg, MD.

Porter, M.F. (1980). An algorithm for suffix stripping. Program, 14(3), 130–137. doi:10.1108/eb046814

Rasolofo, Y., & Savoy, J. (2003). Term proximity scoring for keyword-based retrieval systems. Proceedings of the 25th Annual European Conference on Information Retrieval Research (ECIR '03) (pp. 207–218). Berlin, Germany: Springer-Verlag.

Robertson, S.E., & Walker, S. (1999). Okapi/keenbow at TREC-8. Proceedings of the Eighth Text REtrieval Conference (TREC-8) (pp. 151–161). Washington, DC: GPO.

Seo, J., & Jeon, J. (2009). High precision retrieval using relevance-flow graph. in Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09) (pp. 694–695). New York: ACM Press.

Tao, T., & Zhai, C. (2007). An exploration of proximity measures in information retrieval. Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07) (pp. 295–302). New York: ACM Press.

Troy, A.D., & Zhang, G.-Q. (2007). Enhancing relevance scoring with chronological term rank. Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07) (pp. 599–606). New York: ACM Press.

Weinstein, J.N. (2008). A postgenomic visual icon. Science, 319(5871), 1772–1773. doi:10.1126/science.1151888

Xue, X.-B., & Zhou, Z.-H. (2009). Distributional features for text categorization. IEEE Transactions on Knowledge and Data Engineering, 21(3), 428–442. doi:10.1109/TKDE.2008.166

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. Transactions on Information Systems, 22(2), 179–214. doi:10.1145/984321.984322

Zhao, J., & Yun, Y. (2009). A proximity language model for information retrieval. in Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09) (pp. 291–298). New York: ACM Press.