中国科技论文在线 http://www.paper.edu.cn

Information Processing and Management xxx (2007) xxx–xxx

Towards a unified approach to document similarity search using manifold-ranking of blocks

Xiaojun Wan *, Jianwu Yang, Jianguo Xiao

Institute of Computer Science and Technology, Peking University, Beijing 100871, China

Received 29 April 2007; received in revised form 20 July 2007; accepted 25 July 2007

Abstract

Document similarity search (i.e. query by example) aims to retrieve a ranked list of documents similar to a query document in a text corpus or on the Web. Most existing approaches to similarity search first compute the pairwise similarity score between each document and the query using a retrieval function or similarity measure (e.g. Cosine), and then rank the documents by the similarity scores. In this paper, we propose a novel retrieval approach based on manifold-ranking of document blocks (i.e. a block of coherent text about a subtopic) to re-rank a small set of documents initially retrieved by some existing retrieval function. The proposed approach can make full use of the intrinsic global manifold structure of the document blocks by propagating the ranking scores between the blocks on a weighted graph. First, the TextTiling algorithm and the VIPS algorithm are respectively employed to segment text documents and web pages into blocks. Then, each block is assigned with a ranking score by the manifold-ranking algorithm. Lastly, a document gets its final ranking score by fusing the scores of its blocks. Experimental results on the TDT data and the ODP data demonstrate that the proposed approach can significantly improve the retrieval performances over baseline approaches. Document block is validated to be a better unit than the whole document in the manifold-ranking process.
© 2007 Elsevier Ltd. All rights reserved.

Keywords: Document similarity search; Web similarity search; Manifold-ranking; Document segmentation; Web page segmentation

1. Introduction

Document similarity search (i.e. query by example) is to find documents similar to a query document in a text corpus or on the Web.[1] A ranked list of similar documents is required to be returned to users. The typical kind of similarity search is K-nearest neighbor search, namely K-NN search, which is to find K documents most similar to the query document.

Similarity search is widely used to improve traditional document search engines by allowing the user to use a document as a query and thus releasing the burden of extracting keywords from the document to formulate

* Corresponding author. Tel.: +86 10 82529240; fax: +86 10 82529440.
E-mail address: wanxiaojun@icst.pku.edu.cn (X. Wan).

[1] A document can be either a text document in a text corpus or a web page on the Web.

0306-4573/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2007.07.012

转载

中国科技论文在线 http://www.paper.edu.cn

the query for the user. Traditional search engines take a query of several terms as input and return a set of relevant documents that match the query terms. However, the query terms are sometimes difficult to define and prone to be inaccurate because not every user is a search expert. Note that it happens very often that when users already have an interesting document, they just want to see more relevant documents, and they can use the document as a query and perform similarity search to get similar documents. A few search engines have provided the functionalities of document similarity search or recommendation. For example, *Google*[2] can perform an advanced search with "related" option to find-similar web pages with a user-specified web page and *CiteSeer.IST*[3] provides a list of similar papers with the currently browsed paper. Moreover, document similarity search can be used for relevance feedback in search engines, where the retrieved documents can be re-ranked by their similarity with the documents the user shows interest in.

Different from short-query based search, similarity search usually takes a full document as query and the query is much longer than keyword-based short queries. The query of full document is usually characterized by a series of subtopics and contains more redundant and ambiguous information and even greater noise effects stemmed from the presence of a large number of words unrelated to the overall topic in the document. Document similarity search can be intuitively considered as a particular kind of text retrieval and thus various text retrieval functions can be attempted for this task.

In most previous approaches to similarity search (Cruz, Borisov, Marks, & Webb, 1998; Haveliwala, Gionis, Klein, & Indyk, 2002; Joshi, Agrawal, Krishnapuram, & Negi, 2003; Tombros & Ali, 2005), a document is considered as a single unit for similarity calculation. However, a document as a whole may not be appropriate to represent a single topic because a document usually contains multiple subtopics. Specifically, a text document is usually characterized as a sequence of subtopical discussions (i.e. TextTiles (Hearst, 1994, 1997)) that occur in the context of a few main topic discussions. And a web page usually contains various contents such as navigation, decoration, interaction, contact information, which may be unrelated to the main topic of the web page. Furthermore, a web page often contains multiple topics that are not necessarily relevant to one another. Therefore, detecting the semantic content structure of a document could potentially improve the performance of document similarity search, as well as other standard IR tasks. Moreover, most approaches find relevant documents to a query document only by the pairwise comparison between the document and the query, thus ignoring the intrinsic global manifold structure of the whole set of documents. In order to address the above two limitations, we evaluate document similarity at a finer granularity of document block (or passage) instead of at the coarse granularity of the whole document, each block representing a semantic unit with coherent text reflecting a single topic. And inspired by (Zhou, Weston, Gretton, Bousquet, & Schölkopf, 2003), the manifold-ranking process is employed to make full use of the relationships between the document blocks. The prior assumption of manifold-ranking is: (1) nearby points are likely to have the same ranking scores; (2) points on the same structure (typically referred to as a cluster or a manifold) are likely to have the same ranking scores, similar to the cluster hypothesis in the IR context (Hearst & Pedersen, 1996; van Rijsbergen, 1979). In the manifold-ranking process, document blocks can spread their ranking scores to their nearby neighbors via a weighted network.

In more details, the proposed approach consists of the following two processes: initial ranking and re-ranking. In the initial ranking process, a small number of documents are initially retrieved based on the popular Cosine function. In the re-ranking process, the query document and the initially retrieved documents are segmented into blocks by using either the TextTiling algorithm (Hearst, 1994, 1997) and the VIPS algorithm (Cai, Yu, Wen, & Ma, 2003; Song, Liu, Wen, & Ma, 2004; Yu, Cai, Wen, & Ma, 2003), and then the manifold-ranking algorithm (Zhou, Bousquet, Lal, Weston, & Schölkopf, 2003; Zhou, Weston, et al., 2003) is applied on the blocks and each block obtains its ranking score. Lastly, a document gets its final retrieval score by fusing the ranking scores of the blocks in the document. The initially retrieved documents are re-ranked and the re-ranked list is returned to users. Experimental results on the TDT data and the ODP data show the improved performance of the proposed approach over baseline approaches for similarity search of text documents and

[2] http://www.google.com.
[3] http://citeseer.ist.psu.edu/cs.

中国科技论文在线 http://www.paper.edu.cn

web pages, respectively. Document block is validated to be a more suitable unit than the whole document in the manifold-ranking process.

The main contributions of this paper are two-fold: (1) The manifold-folding algorithm is applied for the first time on document blocks instead of whole documents to improve the retrieval results; (2) The proposed approach is applied not only to text retrieval but also to web retrieval, and the evaluation results on both text documents and web pages show the robustness of the approach.

The rest of this paper is organized as follows: Section 2 introduces related works. The proposed approach is described in detail in Section 3. Sections 4 and 5 give the experiments and results on the TDT data and the ODP data, respectively. Lastly, we present our conclusion and future work in Section 6.

2. Related works

The retrieval performance of a similarity search engine relies heavily on the retrieval function for evaluating document similarity. The popular retrieval functions (including similarity measures) used in current text retrieval systems include the Cosine function, the Jaccard function, the Dice function (Baeza-Yates & Ribeiro-Neto, 1999; van Rijsbergen, 1979), the BM25 function in the Okapi system (Robertson & Walker, 1994; Robertson, Walker, & Beaulieu, 1999), the vector space model with document length normalization in the Smart system (Salton, 1991; Singhal, Buckley, & Mitra, 1996) and the language model (Croft & Lafferty, 2003; Kurland, Lee, & Domshlak, 2005), among which the BM25 function is one of the best models for short-query based search, and the standard Cosine function is considered as one of the best models for document similarity search because of its good ability to measure the similarity between two full documents. More features have been investigated for similarity search of web pages. Haveliwala et al. (2002) propose a new evaluation strategy using Web hierarchies, such as Open Directory, in place of user feedback to evaluate the task of finding pages on the Web that are similar to a query page. Web page structure denoted by HTML tags has also been used to help evaluate web page similarity (Cruz et al., 1998; Joshi et al., 2003). Tombros and Ali (2005) explore the factors of the textual content and the structural information when evaluating web page similarity. Another different source widely-used to determine similarity is the link structure between web pages, such work including (Dean & Henzinger, 1999; Fogaras & Rácz, 2004; Jeh & Widom, 2002; Lin, Lyu, & King, 2006; Xue, Zeng, Chen, & Yu, 2004). Smucker and Allan (2006) examine find-similar, the feature provided by search systems to request documents similar to a given document, as a search tool, like relevance feedback, for improving retrieval performance. In this study, we do not make use of the link structure between web pages.

In recent years, a number of graph-based methods have been proposed to improve text or web retrieval results. For web retrieval, the graph is built based on explicit links between web pages. PageRank (Page, Brin, Motwani, & Winograd, 1998) and HITS (Kleinberg, 1999) are the most popular algorithms that make use of web link structure to improve retrieval performance. PageRank makes use of the link structure between web pages and judge the importance of web pages according to the "votes" or "recommendations" from their neighboring pages. HITS differs from PageRank in that HITS differentiates two kinds of salient web pages: "hub" pages and "authority" pages. For text retrieval, the graph is built based on implicit links inferred from texts. Zhang et al. (2005) propose a novel ranking scheme named Affinity Ranking (AR) to re-rank search results by optimizing both information richness and diversity. The two metrics are calculated using the random walk model and the greedy algorithm, respectively, from a directed link graph named Affinity Graph (AG), which models the structure of a group of documents based on the asymmetric content similarities between each pair of documents. Diaz (2005) exploits the cluster hypothesis (van Rijsbergen, 1979) directly by a score regularization process, which adjusts ad hoc retrieval scores from an initial retrieval so that topically related documents receive similar scores. Kurland and Lee (2005) propose an approach to improving the retrieval precision by adapting the PageRank algorithm for defining and computing centrality within a directed graph and then re-ranking non-hyperlinked document sets. Later, they present an approach (Kurland & Lee, 2006) to utilize cluster information based on the HITS algorithm. The main idea is to perform re-ranking based on centrality within bipartite graphs of documents and clusters, on the premise that these are mutually reinforcing entities. Links between entities are created via consideration of language models induced from them. Qin, Liu, Zhang, Chen, and Ma (2005) propose a generic relevance propagation framework and provide

中国科技论文在线 http://www.paper.edu.cn

a comparison study on the effectiveness and efficiency of various representative propagation models. The above methods have not been explored for document similarity search.

Graph-based methods have also been proposed for document summarization (Erkan & Radev, 2004; Mihalcea & Tarau, 2004) and question answering (Otterbacher, Erkan, & Radev, 2005), where sentences are ranked based on the graph with sentence-to-sentence relationships. Other close related works include passage retrieval (Callan, 1994; Kaszkiel & Zobel, 1997), which ranks passages instead of whole-documents. It can provide convenient units of text to return to the user, avoid the difficulties of comparing documents of different length, and enable identification of short blocks of relevant material amongst otherwise irrelevant text.

3. The proposed approach

3.1. Overview

The aim of the proposed approach is two-fold: one is to evaluate the similarity between the query document and a document at a finer granularity by segmenting the documents into semantic blocks, which addresses the limitation of present similarity metrics based on the whole document, usually characterized as a series of blocks with semantically coherent content; the other is to evaluate the similarity between the query document and a document by exploring the relationships (i.e. the intrinsic manifold structure) between all the obtained blocks in the feature space, which addresses the limitation of present similarity metrics based only on pairwise comparison.

The proposed approach first segments all documents (including the query document) into blocks, and then applies the manifold-ranking process on the document blocks. All the blocks of a document obtain their ranking scores and the ranking score of the document is obtained by fusing the ranking scores of its blocks.

Note that it is of high computational cost to apply the manifold-ranking process to all the documents in the collection, so the above manifold-ranking process is taken as a re-ranking process. First, we use a popular retrieval function (e.g. Cosine) to efficiently obtain an initial ranking of the documents, and then the initial k documents are re-ranked by applying the above manifold-ranking process.

Formally, given a query document q and the collection C, the proposed approach consists of the following four steps:

1. *Initial ranking*: The initial ranking process uses a popular retrieval function (i.e. Cosine) to return an initial ranked list in response to the query document q, and the set of top k documents in the list is denoted as $D_{\text{init}} \subseteq C$, $|D_{\text{init}}| = k$. Each document $d_i \in D_{\text{init}} (1 \leqslant i \leqslant k)$ is associated with an initial retrieval score $InitScore(d_i)$.
2. *Document segmentation*: By using the TextTiling algorithm or the VIPS algorithm, the query document q is segmented into a set of blocks $\chi_q = \{x_1, x_2, \ldots, x_p\}$ and all documents in D_{init} are segmented respectively, and the total set of blocks for D_{init} is $\chi_{D_{\text{init}}} = \{x_{p+1}, x_{p+2}, \ldots, x_n\}$.
3. *Manifold-ranking*: The manifold-ranking process in applied on the whole set of blocks: $\chi = \chi_q \cup \chi_{D_{\text{init}}}$, and each block x_j $(p+1 \leqslant j \leqslant n)$ in $\chi_{D_{\text{init}}}$ gets its ranking score f_j^*.
4. *Score fusion*: The final score $FinalScore(d_i)$ of a document $d_i \in D_{\text{init}}$ $(1 \leqslant i \leqslant k)$ is computed by fusing the ranking scores of its blocks. The documents in D_{init} are re-ranked according to their final scores and the re-ranked list is returned.

The steps 2–4 are key steps in the re-ranking process and they will be illustrated in detail in next sections, respectively.

3.2. The document segmentation process

This step aims to decompose the text documents or web pages into text blocks, with each block representing a single coherent topic. Due to different characteristics of the text documents and web pages, different algorithms are employed for block segmentation, i.e. the TextTiling algorithm (Hearst, 1994, 1997) is employed

中国科技论文在线 http://www.paper.edu.cn

for text segmentation and the VIPS algorithm (Cai et al., 2003; Song et al., 2004; Yu et al., 2003) is employed for web page segmentation. The details of the two algorithms are, respectively, described as follows:

3.2.1. The TextTiling algorithm for text segmentation

There have been several methods for division of text documents according to units such as sections, paragraphs, or fixed length sequences of words, or semantic passages given by inferred shift of topic (Hearst & Plaunt, 1993). In this study, we adopt semantic passages to represent subtopics in a document. As mentioned in (Hearst, 1994, 1997), the text can be characterized as a sequence of subtopical discussions that occur in the context of a few main topic discussions. For example, a news text about China–US relationship, whose main topic is the good bilateral relationship between China and the United States, can be described as consisting of the following subdiscussions (numbers indicate paragraph numbers):

1 Intro-the establishment of China–US relationships.
2–3 The officers exchange visits.
4–5 The culture exchange between the two countries.
6–7 The booming trade between the two countries.
8 Outlook and summary.

We expect to acquire the above subtopics in a document and use them in the manifold-ranking process instead of the whole document. The most popular TextTiling algorithm is used to automatically subdivide text into multi-paragraph units that represent subtopics.

The TextTiling algorithm detects subtopic boundaries by analyzing patterns of lexical connectivity and word distribution. The main idea is that terms that describe a subtopic will co-occur locally, and a switch to a new subtopic will be signaled by the ending of co-occurrence of one set of terms and the beginning of the co-occurrence of a different set of terms. The algorithm has the following three steps:

(1) *Tokenization*: The input text is divided into individual lexical units, i.e. pseudo-sentences of a predefined size;
(2) *Lexical score determination*: All pairs of adjacent lexical units are compared and assigned a similarity value;
(3) *Boundary identification*: The resulting sequence of similarity values is graphed and smoothed, and then is examined for peaks and valleys. The subtopic boundaries are assumed to occur at the largest valleys in the graph.

For TextTiling, subtopic discussions are assumed to occur within the scope of one or more overarching main topics, which span the length of the text. Since the segments are adjacent and non-overlapping, they are called TextTiles. In this study, we denote a TextTile as a block.

The computational complexity is approximately linear with the document length, and a few efficient implementations are available, such as Kaufmann (Kaufmann, 1999) and JTextTile (Choi, 1999).

3.2.2. The VIPS algorithm for page segmentation

Several kinds of methods have been proposed for web page segmentation, among which the most popular ones are DOM-based segmentation (Chen, Zhou, Shi, Zhang, & Qiu, 2001), location-based segmentation (Kovacevic, Diligenti, Gori, & Milutinovic, 2002) and Vision-based Page Segmentation (VIPS) (Cai et al., 2003; Song et al., 2004; Yu et al., 2003). Compared with other segmentation algorithms, VIPS excels in both an appropriate partition granularity and coherent semantic aggregation. In this study, we adopt the VIPS algorithm to segment web pages into semantic blocks. Note that other segmentation algorithms can also be explored as well, which is however not the focus of this study.

The VIPS algorithm makes full use of page layout features such as font, color and size and takes advantage of visual cues to obtain the vision-based content structure of a web page. The algorithm can successfully bridge the gap between the DOM structure and the semantic structure. The page is partitioned based on visual

中国科技论文在线 http://www.paper.edu.cn

separators and structured as a hierarchy closely related to how a user will browse the page. Content related parts could be grouped together even if they are in different branches of the DOM tree.

The VIPS algorithm first extracts all the suitable nodes from the HTML DOM tree, and then finds the separators between these nodes. Here, separators denote the horizontal or vertical lines in a web page that visually do not cross any node. Based on these separators, the semantic tree of the web page is constructed. A value called degree of coherence (DoC) is assigned for each node to indicate how coherent it is. Consequently, VIPS can efficiently keep related content together while separating semantically different blocks from each other.

Each block in VIPS is represented as a node in a tree. The root is the whole page; inner nodes are the top level coarser blocks, and all leaf nodes consist of a flat segmentation of a web page. The granularity of segmentation in VIPS is controlled by a predefined degree of coherence (PDoC), which plays a role as a threshold of the most appropriate granularity for different applications (usually set to 5). The segmentation only stops when the DoCs of all blocks are no smaller than the PDoCs. Fig. 1 shows the result of using VIPS to segment a sample CNN web page (Song et al., 2004).

Fig. 1. VIPS segmentation of a sample web page.

中国科技论文在线 http://www.paper.edu.cn

VIPS is also very efficient. Since we trace down the DOM structure for visual block extraction and do not analyze every basic DOM node, the algorithm is totally top-down. Furthermore, the PDoC can be predefined, which brings significant flexibility to segmentation and greatly improve the performance.

3.3. The manifold-ranking process

Manifold-ranking (Zhou, Bousquet, et al., 2003; Zhou, Weston, et al., 2003) is a universal ranking algorithm initially used to rank data points along their underlying manifold structure. Text retrieval experiments using manifold-ranking of documents have been performed and promising results have been obtained on the 20-newsgroups dataset (Zhou, Weston, et al., 2003). An intuitive description of manifold-ranking is as follows: A weighted network is formed on the data, and a positive rank score is assigned to each known relevant point and zero to the remaining points which are to be ranked. All points then spread their ranking score to their nearby neighbors via the weighted network. The spread process is repeated until a global stable state is achieved, and all points obtain their final ranking scores.

In our context, the data points are denoted by the document blocks in the query document q and the documents in D_{init}. The manifold-ranking process in our context can be formalized as follows:

Given a set of data points $\chi = \chi_q \cup \chi_{D_{\text{init}}} = \{x_1, x_2, \ldots, x_p, x_{p+1}, \ldots, x_n\} \subset R^m$, the first p points are the document blocks in the query document q and the rest $n-p$ points are the document blocks in the documents in D_{init}. Let $f: \chi \to R$ denote a ranking function which assigns to each point $x_j (1 \leqslant j \leqslant n)$ a ranking value f_j. We can view f as a vector $\boldsymbol{f} = [f_1, f_2, \ldots, f_n]^{\mathrm{T}}$. We also define a vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]^{\mathrm{T}}$, in which $y_j = 1$ $(1 \leqslant j \leqslant p)$ for the document blocks in q and $y_j = \mathit{InitScore}(d_i) (p+1 \leqslant j \leqslant n)$ for the document blocks in any document d_i in D_{init}, where $x_j \in d_i$, $d_i \in D_{\text{init}}$, and $\mathit{InitScore}(d_i) \in [0,1]$, which means that the initial retrieval score of a document is used as the initial ranking scores of the blocks in the document. The manifold-ranking algorithm goes as in Fig. 2.

In the above iterative algorithm, the normalization in the third step is necessary to prove the algorithm's convergence. The fourth step is the key step of the algorithm, where all points spread their ranking score to their neighbors via the weighted network. The parameter of manifold-ranking weight α specifies the relative contributions to the ranking scores from neighbors and the initial ranking scores. Note that *self-reinforcement* is avoided since the diagonal elements of the affinity matrix are set to zero.

The theorem in (Zhou, Weston, et al., 2003) guarantees that the sequence $\{f(t)\}$ converges to

$$\boldsymbol{f}^* = \beta(\boldsymbol{I} - \alpha \boldsymbol{S})^{-1} \boldsymbol{y} \tag{1}$$

where $\beta = 1 - \alpha$. Although $\boldsymbol{f}^*$ can be expressed in a closed form, for large scale problems, the iteration algorithm is preferable due to computational efficiency. Usually the convergence of the iteration algorithm is achieved when the difference between the scores computed at two successive iterations for any point falls below a given threshold (0.0001 in this study).

Using Taylor expansion, we have

$$\boldsymbol{f}^* = \beta(\boldsymbol{I} - \alpha \boldsymbol{S})^{-1} \boldsymbol{y} = \beta(\boldsymbol{I} + \alpha \boldsymbol{S} + \alpha^2 \boldsymbol{S}^2 + K)\boldsymbol{y} = \beta(\boldsymbol{y} + \alpha \boldsymbol{S}\boldsymbol{y} + \alpha \boldsymbol{S}(\alpha \boldsymbol{S}\boldsymbol{y}) + K) \tag{2}$$

From the above equation, if we omit the constant coefficient β, $\boldsymbol{f}^*$ can be regarded as the sum of a series of infinite terms. The first term is simply the vector $\boldsymbol{y}$, and the second term is to spread the ranking scores of the

1. *Compute the pairwise similarity among points (document blocks) using the standard Cosine function.*
2. *Connect any two points with an edge. We define the affinity matrix* $\boldsymbol{W}$ *by* $W_{ij} = sim_{cosine}(x_i, x_j)$, *where* $sim_{cosine}(x_i, x_j)$, *is the Cosine similarity value between* x_i *and* x_j. *Note that we let* $W_{ii} = 0$ *to avoid loops in the graph built in next step.*
3. *Symmetrically normalize* $\boldsymbol{W}$ *by* $\boldsymbol{S} = \boldsymbol{D}^{-1/2} \boldsymbol{W} \boldsymbol{D}^{-1/2}$ *in which* $\boldsymbol{D}$ *is the diagonal matrix with (i,i)-element equal to the sum of the i-th row of* $\boldsymbol{W}$.
4. *Iterate* $\boldsymbol{f}(t+1) = \alpha \boldsymbol{S}\boldsymbol{f}(t) + (1-\alpha)\boldsymbol{y}$ *until convergence, where* α *is a parameter in [0,1).*
5. *Let* f_j^* *denote the limit of the sequence* $\{f_j(t)\}$. *Each block* x_j $(p+1 \leqslant j \leqslant n)$ *gets its ranking score* f_j^*.

Fig. 2. The manifold-ranking algorithm.

中国科技论文在线 http://www.paper.edu.cn

blocks to their nearby blocks, and the third term is to further spread the ranking scores, etc. Thus the effect of the blocks in the documents is gradually incorporated into the ranking score.

3.4. The score fusion process

The final retrieval score of a document $d_i \in D_{\text{init}}$ is computed by fusing the ranking scores of its blocks as follows:

$$FinalScore(d_i) = \frac{\sum_{x_j \in d_i} \lambda_j f_j^*}{|d_i|} \tag{3}$$

where λ_j measures the importance of the block x_j in document d_i. The bigger λ_j is, the more important the block is. $|d_i|$ represents the number of blocks in document d_i. This normalization avoids favoring documents with more blocks.

For text documents, $\lambda_j = \text{sim}_{\text{cosine}}(x_j, d_i)$ is the Cosine similarity between the block (i.e. TextTile) x_j and its associated document d_i, which measures the importance of the block (i.e. TextTile) x_j in the document d_i.

For web pages, $\lambda_j = \sigma \frac{\text{size of block } x_j \text{ in page } d_i}{\text{distance from the center of } x_j \text{ to the center of screen}}$ empirically measures the importance of the block x_j in the web page d_i (Cai, He, Wen, & Ma, 2004), where σ is a normalization factor to make the sum of λ_j for blocks in d_i to be 1, i.e. $\sum_{x_j \in d_i} \lambda_j = 1$.

Finally, the documents in D_{init} are re-ranked according to their final scores and the re-ranked list is returned. Note that only the top k documents (i.e. the documents in D_{init}) in the initial ranked list are re-ranked and the rest documents in the initial ranked list still hold their initial ranks.

4. Evaluation on TDT data

This section aims to evaluate the proposed approach to similarity search of text documents on a text corpus.

4.1. Experimental setup

4.1.1. Baseline approach

In the experiments, the manifold-ranking based approach (“MR + TextTile”) is compared with the following baseline approaches: “BM25”, “NVSM”, “Cosine” and “MR + Document”. The “BM25”, “NVSM” and “Cosine” baselines do not apply the manifold-ranking process and directly ranks the documents by their similarity with the query document based on respective similarity functions. The “MR + Document” baseline is adopted in (Zhou, Weston, et al., 2003), which uses the whole document instead of TextTile in the manifold-ranking process, and thus it does not need the steps of text segmentation and score fusion. The manifold-ranking process is also applied on top-retrieved documents by other retrieval functions.

The Cosine function is one of the most popular measures for evaluating document similarity and it is based on the vector space model (VSM) (Baeza-Yates & Ribeiro-Neto, 1999; Salton, Wong, & Yang, 1975). Each document d is represented by a vector with each dimension referring to a unique term. The weight $w_{d,t}$ associated with term t is calculated by the $tf_{d,t}.idf_t$ formula, where $tf_{d,t}$ is the number of occurrences of term t in document d and $idf_t = 1 + \log(N/n_t)$ is the inverse document frequency, where N is the total number of documents in the collection and n_t is the number of documents containing term t. The Cosine similarity $sim(q,d)$, between the query document q and any document d, can be defined as the normalized inner product of the two corresponding vectors as follows:

$$\text{sim}_{\text{Cosine}}(q,d) = \frac{\sum_{t \in q \cap d} (w_{q,t} \cdot w_{d,t})}{\sqrt{\sum_{t \in q} w_{q,t}^2 \times \sum_{t \in d} w_{d,t}^2}} \tag{4}$$

中国科技论文在线 http://www.paper.edu.cn

The BM25 function (Robertson & Walker, 1994; Robertson et al., 1999) is one of the most popular retrieval models in a probabilistic framework and is widely-used in the Okapi system. Given the query document q, the similarity score for document d is defined as follows[4]:

$$\mathrm{sim}_{\mathrm{BM25}}(q,d)=\sum_{t\in q} tf_{q,t}\times\log\left(\frac{N-n_t+0.5}{n_t+0.5}\right)\times\frac{(K+1)\times tf_{d,t}}{K\times\left\{(1-b)+b\frac{dltf_d}{avedltf}\right\}+tf_{d,t}} \tag{5}$$

where $dltf_d$ is the sum of term frequencies in d; $avedltf$ is the average of $dltf_d$ in the collection; The parameters are tuned on a training corpus and they are set as follows: $K=2.0$, $b=0.8$.

The vector space model with document length normalization (NVSM) (Salton, 1991; Singhal et al., 1996) is also a popular retrieval model and is used in the Smart system (Salton, 1991). Given the query document q, the similarity score for document d is defined as follows:

$$\mathrm{sim}_{\mathrm{NVSM}}(q,d)=\sum_{t\in q}(1+\log(tf_{q,t}))\times idf_t\times\frac{1+\log(tf_{d,t})}{1+\log(avetf_d)}\times\frac{1}{avedlb+S\times(dlb_d-avedlb)} \tag{6}$$

where dlb_d is the number of unique terms in d; $avetf_d$ is the average of term frequencies in d (i.e. " $dltf_d/dlb_d$"); $avedlb$ is the average of dlb_d in the collection; S is tuned and set to 0.2.[5]

For "MR + Document" and "MR + TextTile", the Cosine function is heuristically used for initial ranking and manifold-ranking. Note that other similarity measures can also be explored, but this study focuses only on the widely-used Cosine measure.

4.1.2. Dataset

A ground truth dataset is required to perform the experiments. We built the ground truth dataset from the TDT-3 corpus, which has been used for evaluation of the task of topic detection and tracking (Allan, Carbonell, Doddington, Yamron, & Yang, 1998) in 1999 and 2000. TDT-3 corpus is annotated by Linguistic Data Consortium (LDC) from 8 English sources and 3 Mandarin sources for the period of October through December 1998. 120 topics are defined and about 9000 stories are annotated over these topics with an "on-topic" table presenting all stories explicitly marked as relevant to a given topic. According to the specification of TDT, the on-topic stories within the same topic are similar and relevant. After removing the stories written in Chinese, there remain 8458 English stories. We randomly chose 40 topics as a test set, while the others were used as a training set for parameter tuning.[6]

Sentence tokenization was firstly applied to all documents. Stop words were removed and Porter's stemmer (Porter, 1980) was used for word stemming. The JTextTile tool (Choi, 1999) with default setting was employed to segment each document into TextTiles. The total stories are used as the document collection for search, the first document within the topic is simply used as the query document without any tuning, and all the other documents within the same topic are the relevant (similar) documents, while all the documents within other topics are considered irrelevant (dissimilar) to the query document. A ranked list of 500 documents was required to be returned for each query document based on each retrieval approach. The higher the document is in the ranked list, the more similar it is with the query document. For the proposed manifold-ranking process, the number of re-ranked documents is heuristically set to 50, i.e. $|D_{\mathrm{init}}|=k=50$.

4.1.3. Evaluation metric

As in TRECT[7] experiments, we use the average precisions at top N results, i.e. $P@5$ and $P@10$, and the mean average precision (MAP) as evaluation metrics.

The precision at top N results for a query is calculated as follows:

[4] The BM25 and NVSM formulae are defined according to (Iwayama, Fujii, Kando, & Marukawa, 2003).

[5] The parameter value was varied on the training set and then the value corresponding to the highest performance (MAP) was chosen and used on the test set.

[6] The parameters were optimized on the training set with respect to the MAP metric defined in next section.

[7] http://trec.nist.gov.

中国科技论文在线 http://www.paper.edu.cn

$$P@N = \frac{|C \cap R|}{|R|}, \tag{7}$$

where R is the set of top N retrieved documents, and C is the set of similar documents defined above for a given query document. The precision is calculated for each query and then the values are averaged across all queries.

The non-interpolated average precision (AP) for a query is a number averaged over all precision values calculated after each relevant document is retrieved. If a relevant document is not retrieved, the corresponding precision value is 0.0. The mean average precision (MAP) is the mean AP over all queries.

Note that the number of documents within each topic is different and some topics contain even less than 5 documents or 10 documents, so its corresponding $P@5$ or $P@10$ may be low.

4.2. Experimental results

The precision values of the proposed approach (''MR + TextTile'') and the baseline approaches (i.e. ''Cosine'' and ''MR + Document'') are compared in Table 1, where the manifold-ranking weight α is set to 0.3, which is tuned on the training set. Seen from Table 1, the proposed approach significantly outperforms all the baseline systems over $P@5$ and $P@10$ metrics. We can also see that the ''MR + Document'' baseline achieves almost the same $P@5$ value with the ''Cosine'' baseline and the higher $P@10$ and MAP values than the ''Cosine'' baseline, which demonstrates that manifold-ranking process can benefit document ranking.

Among the three baselines without using manifold-ranking (i.e. ''BM25'', ''NVSM'', ''Cosine''), the BM25 function and the NVSM function perform poorly, while the Cosine function achieves a comparatively high performance, especially over the MAP metric. Note that both the BM25 function and the NVSM function work well for short-query based text retrieval. The result shows that measuring the similarity between full documents is different from measuring the similarity between a short query and a full document (i.e. the keyword search in TREC experiments). The query of a full document is different from the relatively short query in that the full document contains more redundant and ambiguous information and even greater noise effects stemming from the presence of a large number of words unrelated to the overall topic in the document, and thus measuring the similarity between full documents and measuring the similarity between the short query and a full document are not identical.

The performances of the two MR-based approaches (i.e. ''MR + TextTile'' & ''MR + Document'') with different manifold-ranking weight α are shown and compared in Figs. 3–5. Seen from the figures, with appropriate values of the manifold-ranking weight (i.e. $\alpha < 0.5$), the proposed approach (i.e. ''MR + TextTile'') can significantly outperform the approach of ''MR + Document'' over $P@5$ and $P@10$ metrics (t-test: p-value < 0.05), which demonstrates that TextTile is a more appropriate unit than the whole document for the manifold-ranking process. This result can be explained by that a document is usually characterized as a sequence of subtopical discussions that occur in the context of a few main topic discussions and each TextTile can represent a subtopic with coherent text, and thus the manifold-ranking process can work at a finer granularity.

We now compare the performances of the proposed approaches (''MR + TextTile'') with manifold-ranking ($\alpha = 0.3$) and without manifold-ranking ($\alpha = 0$). Seen from the figures, the $P@5$ and $P@10$ values of ''MR + TextTile'' with $\alpha = 0.3$ are slightly better than those of ''MR + TextTile'' with $\alpha = 0$, while the

Table 1
Performance comparison of the proposed approach and baseline approaches on TDT data

	BM25	NVSM	Cosine	MR + Document	MR + TextTile
$P@5$	0.820	0.810	0.830[c]	0.825	**0.855**[a,b,c]
$P@10$	0.720	0.710	0.720	0.738[a,c]	**0.763**[a,b,c]
MAP	0.757	0.723	0.820[c]	0.823[c]	**0.833**[c]

[a] Performance change over ''Cosine'' is statistically significant.
[b] Performance change over ''MR + Document'' is statistically significant.
[c] Performance changes over ''BM25'' and ''NVSM'' are both statistically significant; ''statistically significant'' means p-value < 0.05 for t-test.

中国科技论文在线 http://www.paper.edu.cn

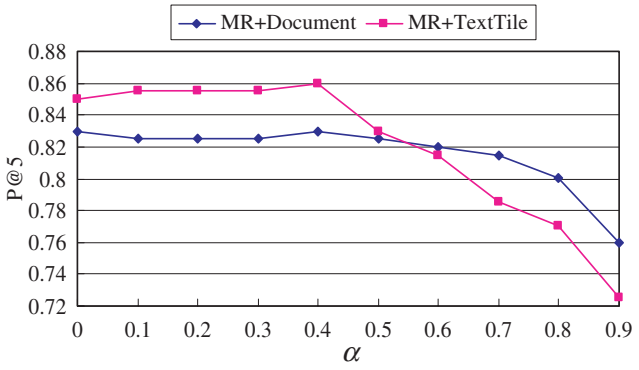

Fig. 3. *P*@5 comparison of MR-based approaches with different α (TDT data).

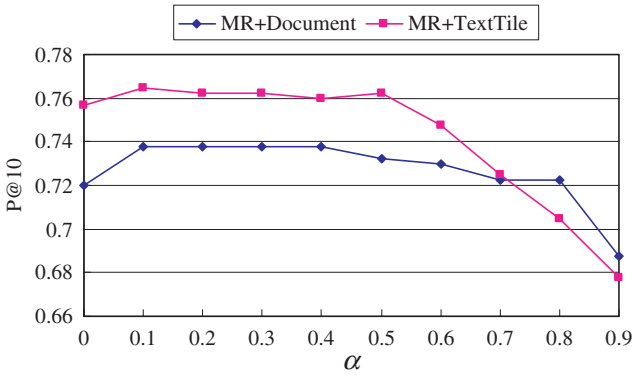

Fig. 4. *P*@10 comparison of MR-based approaches with different α (TDT data).

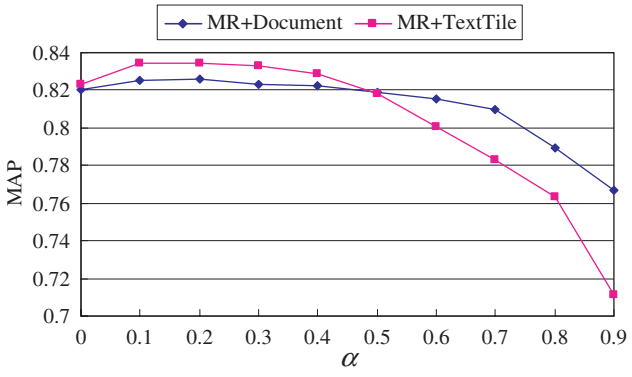

Fig. 5. MAP comparison of MR-based approaches with different α (TDT data).

MAP value of "MR + TextTile" with $\alpha = 0.3$ is significantly better than that of "MR + TextTile" with $\alpha = 0$ (*t*-test: *p*-value < 0.05). Note that "MR + TextTile" with $\alpha = 0$ can be considered as a basic passage-based re-ranking approach. The results show that in addition to the contribution of using TextTiles instead of whole documents, the manifold-ranking process does contribute to the final retrieval performance.

Fig. 6 explores the influence of the number of initially retrieved documents (i.e. *k*) on the performance of the proposed approach (i.e. "MR + TextTile"). Seen from the figure, when *k* is larger than 75, the system performances almost do not alter any more. This shows that a small number of initially retrieved documents work

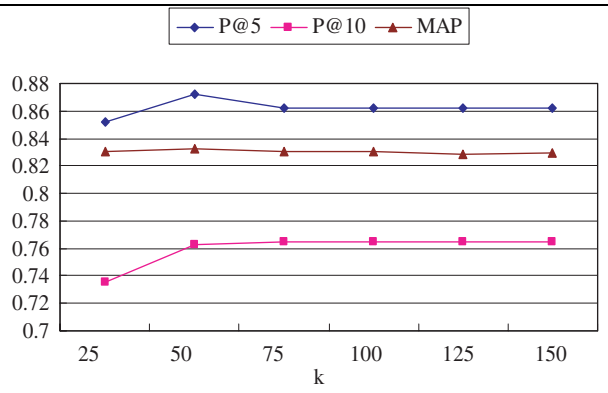

Fig. 6. Performance comparison of the proposed approach with different k (TDT data).

well in the re-ranking process and it does not improve the retrieval performance by increasing the number of initially retrieved documents for re-ranking. In the TDT dataset, the number of the relevant (similar) documents for a query document is not large, and the top 50 documents in the initial ranked list usually include most relevant documents. Most irrelevant documents are likely to be added into the re-ranked document set when k is increased, which does not benefit to improve the retrieval performance at all.

5. Evaluation on ODP data

This section aims to evaluate the proposed approach to similarity search of web pages on the Web.

5.1. Experimental Setup

Similar to the experiments in Section 4.1, the proposed approach (“MR + Block”) is compared with the following baseline approaches: “BM25”, “NVSM”, “Cosine” and “MR + Page”. The “MR + Page” baseline uses the whole web page instead of page block in the manifold-ranking process, and thus it does not apply the steps of page segmentation and score fusion as in the proposed approach. For “MR + Page” and “MR + Block”, the Cosine function is heuristically used for initial ranking and manifold-ranking.

As in (Haveliwala et al., 2002), we built the ground truth dataset from the directory hierarchy of Open Directory Project (ODP).[8] The ODP maintains hierarchical directories with a large number of web pages, and all the web pages within a directory belong to the same category. Document similarity is implicitly encoded in these hierarchical directories, e.g. for a web page in the directory of “\Computers\Programming\Databases”, another web page in the same directory is assumed to be more similar to this page than any web page in the directory of “\Computers\Programming\Graphics”. We downloaded 9610 web pages from subdirectories of five top directories: “\Business”, “\Computers”, “\Recreation”, “\Science” and “\Sports”. Each web page belonged to a three-level directory, e.g. “\Science\Agriculture\Animals”. There are on average about 25 web pages in each directory. We used the VIPS tool[9] for page segmentation. The average number of blocks in a web page is 6.59. We extracted plain texts from the whole web page and each block by removing all tags. Stop words were removed and Porter’s stemmer was used for word stemming. 170 web pages were randomly collected as queries in test set. Two kinds of relevance are defined as follows:

1-level relevance: Given a query page q in the directory of “\A\B\C”, only the web pages in the same directory of “\A\B\C” are considered as relevant (similar) pages, and all other pages are considered as irrelevant (dissimilar) pages.

2-level relevance: Given a query page q in the directory of “\A\B\C”, the web pages in all the subdirectories of “\A\B” (e.g. “\A\B\D”) are considered as relevant (similar) pages, and all other pages are considered as irrelevant (dissimilar) pages.

[8] http://dmoz.org.
[9] http://www.ews.uiuc.edu/~dengcai2/VIPS/VIPS.html.

Based on the above relevance levels, two relevance lists for a query page were built, respectively, and the retrieval performance can be obtained on either relevance list.

The total web pages were used as the collection for search, a ranked list of 500 web pages was returned in response to each query page based on a specified retrieval approach. For the proposed manifold-ranking process, the number of re-ranked web pages was set to 50, i.e. $|D_{\text{init}}| = k = 50$.

Similarly, we used *P*@5, *P*@10 and MAP as evaluation metrics. Note that based on different relevance lists, we can obtain respective *P*@5, *P*@10 and MAP values. Usually the precision values based on the 2-level relevance list are higher than those based on the 1-level relevance list.

5.2. Experimental results

The precision values of the proposed approach ("MR + Block") and the baseline approaches are compared in Table 2, where the manifold-ranking weight α is also set to 0.3, the same value with the evaluation on the TDT dataset. Seen from Table 2, the proposed approach significantly outperforms the baseline approaches. We can also see that the "MR + Page" baseline achieves higher precision values than the "Cosine" baseline based on 2-level relevance, while it achieves lower precision values than the "Cosine" baseline based on 1-level relevance, which shows that the manifold-ranking process on the granularity of the whole web page can not significantly improve the retrieval performance. Similarly, the Cosine function outperforms the BM25 function and the NVSM function over all metrics, which further demonstrates that document similarity search is different from short-query based search.

The performance values of two MR-based approaches (i.e. "MR + Block" & "MR + Page") with different manifold-ranking weight α are shown and compared in Figs. 7–12. Seen from the figures, with appropriate

Table 2
Precision values of the proposed approach and baseline approaches on ODP data

	BM25	NVSM	Cosine	MR + Page	MR + Block
1-level relevance					
P@5	0.484	0.480	0.515[c]	0.504[c]	**0.524**[c]
P@10	0.357	0.369	0.386[c]	0.383	**0.405**[a,b,c]
MAP	0.370	0.360	0.414[c]	0.414[c]	**0.420**[c]
2-level relevance					
P@5	0.660	0.671	0.689[c]	0.698	**0.718**[a,c]
P@10	0.570	0.572	0.599	0.617[a,c]	**0.640**[a,b,c]
MAP	0.534	0.538	0.580[c]	0.618[a,c]	**0.625**[a,c]

[a] Performance change over "Cosine" is statistically significant.
[b] Performance change over "MR + Page" is statistically significant.
[c] Performance changes over "BM25" and "NVSM" are both statistically significant.

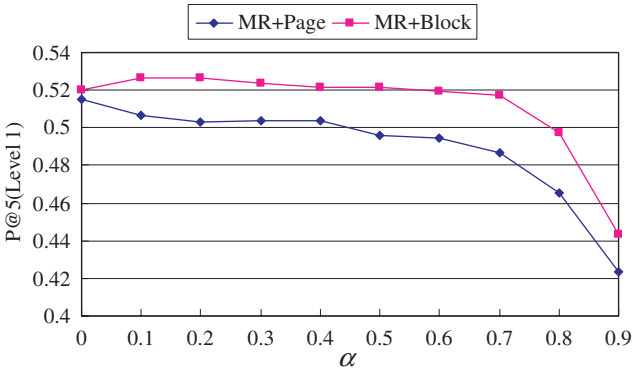

Fig. 7. *P*@5 comparison of MR-based approaches with different manifold-ranking weights based on 1-level relevance (ODP data).

values of the manifold-ranking weight ($\alpha < 0.5$), the proposed approach (i.e. "MR + Block") can always outperform the approach of "MR + Page". The observations demonstrate that page block is a more appropriate unit than the whole web page in the manifold-ranking process, which can be explained by that a web page usually contains various contents and multiple topics and is not appropriate to be considered as a single unit, while each block is assumed to represent a single topic with coherent text and thus is a finer granularity for the manifold-ranking process.

We now compare the performances of the proposed approaches ("MR + Block") with manifold-ranking ($\alpha = 0.3$) and without manifold-ranking ($\alpha = 0$). Seen from Figs. 7–9, the *P*@5, *P*@10 and MAP values (based

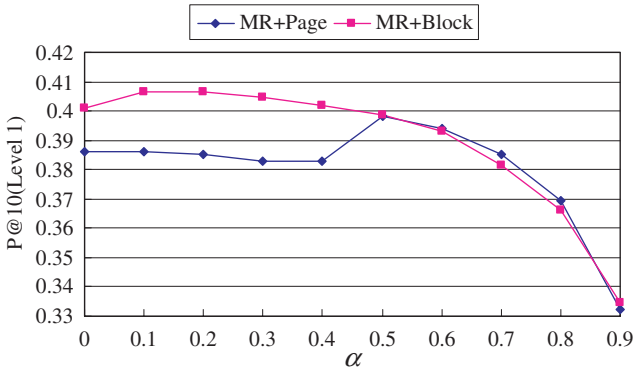

Fig. 8. *P*@10 comparison of MR-based approaches with different manifold-ranking weights based on 1-level relevance (ODP data).

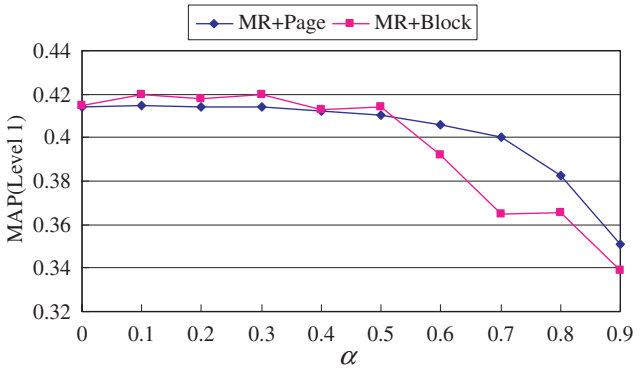

Fig. 9. MAP comparison of MR-based approaches with different manifold-ranking weights based on 1-level relevance (ODP data).

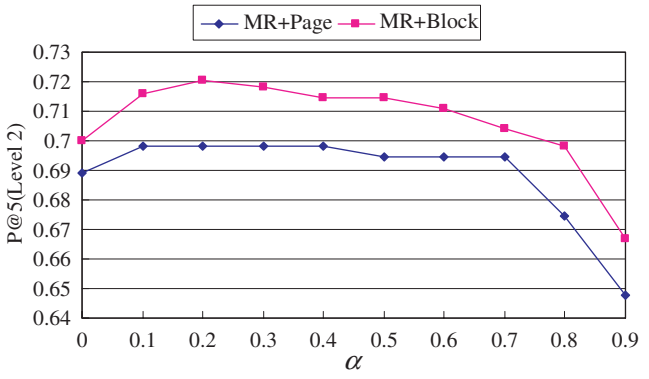

Fig. 10. *P*@5 comparison of MR-based approaches with different manifold-ranking weights based on 2-level relevance (ODP data).

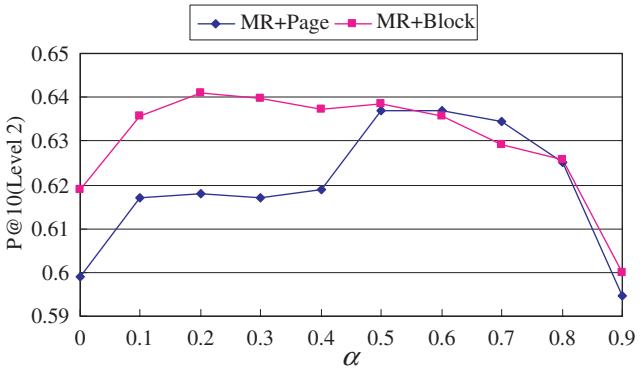

Fig. 11. *P*@10 comparison of MR-based approaches with different manifold-ranking weights based on 2-level relevance (ODP data).

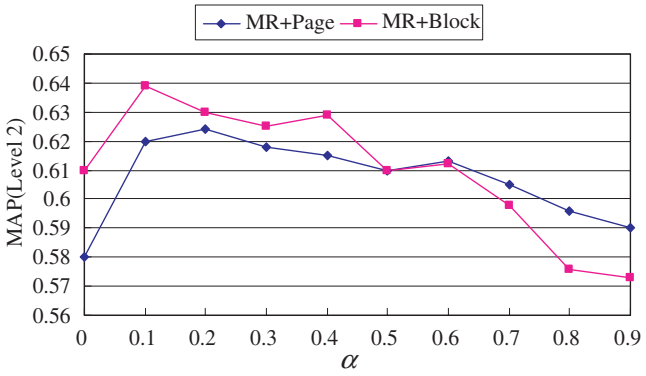

Fig. 12. MAP comparison of MR-based approaches with different manifold-ranking weights based on 2-level relevance (ODP data).

on 1-level relevance) of "MR + Block" with $\alpha = 0.3$ are slightly better than those of "MR + Block" with $\alpha = 0$, while seen from Figs. 10–12, the *P*@5, *P*@10 and MAP values (based on 2-level relevance) of "MR + Block" with $\alpha = 0.3$ are all significantly better than those of "MR + Block" with $\alpha = 0$ (*t*-test: *p*-value < 0.05). The results show that in addition to the contribution of using page blocks instead of whole web pages, the manifold-ranking process does contribute to the final retrieval performance.

Fig. 13 explores the influence of the number of initially retrieved documents (i.e. *k*) on the performances of our proposed approach (i.e. "MR + Block"). Seen from the figure, when *k* increases from 50 to 150, the

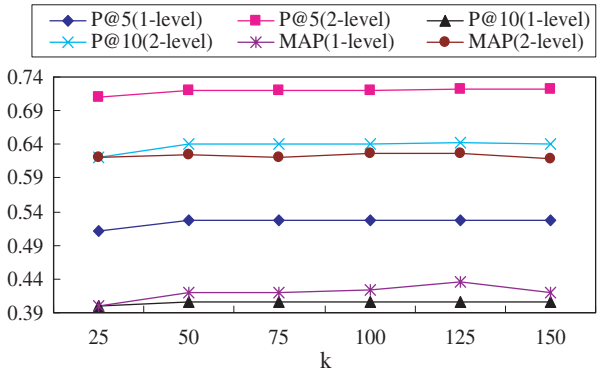

Fig. 13. Performance comparison of the proposed approach with different number of initially retrieved documents (ODP data).

中国科技论文在线 http://www.paper.edu.cn

performances almost do not alter any more. This shows that a small number of initially retrieved documents work well in the re-ranking process and it will not significantly improve the retrieval performance by increasing the number of initially retrieved documents, similar to the results on the TDT dataset.

6. Conclusion and future work

In this paper, we propose a novel retrieval approach for document similarity search. The proposed approach re-ranks a small number of initially retrieved documents based on manifold-ranking of document blocks. The manifold-ranking process can make use of the relationships among document blocks to improve the retrieval performance. Experiments on the TDT data and the ODP data have been performed separately and the experimental results demonstrate the favorable performance of the proposed approach to both similarity search of text documents and similarity search of web pages.

In this study, we employ the TextTiling algorithm for text segmentation and use the VIPS algorithm for web page segmentation. We will investigate how different document segmentation methods affect the retrieval performance in future work. Furthermore, more diversified data sets will be used in the experiments to thoroughly investigate the robustness of the proposed approach. Retrieval efficiency is important for real search engines, so we will explore faster algorithms to propagating the ranking scores between document blocks.

References

Allan, J., Carbonell, J., Doddington, G., Yamron, J. P. & Yang, Y. (1998). Topic detection and tracking pilot study: Final report. In *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop* (pp. 194–218).

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrival*. ACM Press and Addison Wesley.

Cai, D., Yu, S., Wen, J. -R., & Ma, W. -Y. (2003). VIPS: A vision based page segmentation algorithm. *Microsoft technical report*, MSR-TR-2003-79.

Cai, D., He, X., Wen, J. -R., & Ma, W. -Y. (2004). Block-level link analysis. In *Proceedings of the 27th annual international ACM SIGIR conference (SIGIR'2004)*.

Callan, J. (1994). Passage-level evidence in document retrieval, In *Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 302–310).

Chen, J., Zhou, B., Shi, J., Zhang, H. -J., & Qiu, F. (2001). Function-based object model towards website adaptation. In *Proceedings of the 10th world wide web conference (WWW10)*.

Choi, F. (1999). JTextTile: A free platform independent text segmentation algorithm. http://www.cs.man.ac.uk/~choif.

Croft, B., & Lafferty, J. (2003). *Language modeling for information retrieval*. Kluwer Academic Publishers.

Cruz, I. F., Borisov, S., Marks, M. A., Webb, T. R. (1998). Measuring structural similarity among web documents: Preliminary results. In *Proceedings of the 7th international conference on electronic publishing* (pp. 513–524).

Dean, J., & Henzinger, M. R. (1999). Finding related pages in the World Wide Web. In *Proceedings of the eighth international conference on world wide web* (pp. 1467–1479).

Diaz, F. (2005). Regularizing ad hoc retrieval scores. In *Proceedings of the 14th ACM international conference on information and knowledge management (CIKM'2005)*.

Erkan, G., & Radev, D. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR), 22*, 457–479.

Fogaras, D., & Rácz, B. (2004). Scaling link-based similarity search. *Technical report*.

Haveliwala, T.H., Gionis, A., Klein, D., Indyk, P. (2002). Evaluating strategies for similarity search on the Web. In *Proceedings of WWW2002* (pp. 432–442).

Hearst, M. A. (1994). Multi-paragraph segmentation of expository text. In *Proceedings of the 32P^{ndP} meeting of the association for computational linguistics*, Los Cruces, NM.

Hearst, M. A. (1997). TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics, 23*(1), 33–64.

Hearst, M. A., & Pedersen, O. (1996). Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of SIGIR1996* (pp. 76–84).

Hearst, M. A., & Plaunt, C. (1993). Subtopic structuring for full-length document access. In *Proceedings of the 16P^{thP} annual international ACM/SIGIR conference*, Pittsburgh, PA.

Iwayama, M., Fujii, A., Kando, N., Marukawa, Y. (2003). An empirical study on retrieval models for different document genres: Patents and newspaper articles. In *Proceedings of SIGIR2003*.

Jeh, G., & Widom, J. (2002). SimRank: A measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining*.

Joshi, S., Agrawal, N., Krishnapuram, R., Negi, S. (2003). A bag of paths model for measuring structural similarity in web documents. In *Proceedings of the 9th ACM SIGKDD conference* (pp. 577–582).

Kaszkiel, M., & Zobel, J. (1997). Passage retrieval revisited. *ACM SIGIR Forum, 31*(SI), 178–185.

中国科技论文在线 http://www.paper.edu.cn

Kaufmann, S. (1999). Cohesion and collocation: Using context vectors in text segmentation, In *Proceedings of the 37th conference on association for computational linguistics* (pp. 591–595).

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM, 46*(5), 604–632.

Kovacevic, M., Diligenti, M., Gori, M., & Milutinovic, V. (2002). Recognition of common areas in a web page using visual information: A possible application in a page classification. In *Proceedings of 2002 IEEE international conference on data mining (ICDM'02)*, Maebashi City, Japan.

Kurland, O., Lee, L. (2005). PageRank without hyperlinks: Structural re-ranking using links induced by language models. In *Proceedings of SIGIR2005* (pp. 306–313).

Kurland, O., Lee, L. (2006). Respect my authority! HITS without hyperlinks: utilizing cluster-based language models. In Proceedings of SIGIR2006.

Kurland, O., Lee, L., Domshlak, C. (2005). Better than the real thing? Iterative pseudo-query processing using cluster-based language models. In *Proceedings of SIGIR2005*.

Lin, Z., Lyu, M.R., King, I. (2006). PageSim: A novel link-based measure of web page similarity. In *Proceeding of the 15th international world wide web conference*.

Mihalcea, R., Tarau, P. (2004). Textrank: Bringing order into texts. In *Proceedings of EMNLP 2004* (pp. 404–411).

Otterbacher, J., Erkan, G., Radev, D. (2005). Using random walks for question-focused sentence retrieval. In *Proc. HLT/EMNLP 2005*.

Page, L., Brin, S., Motwani, R., Winograd, T. (1998). The pagerank citation ranking: Bringing order to the web, *Technical report*. Stanford, CA: Stanford University.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137.

Qin, T., Liu, T. -Y., Zhang, X. -D., Chen, Z., Ma, W. -Y. (2005). A study of relevance propagation for web search. In *Proceedings of SIGIR2005*.

Robertson, S., Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proc. of the 17th international ACM/SIGIR conference on research and development in information retrieval* (pp. 232–241).

Robertson, S., Walker, S., Beaulieu, M. (1999) Okapi at TREC-7: Automatic ad hoc, filtering, VLC and filtering tracks. In *Proceedings of TREC'99*.

Salton, G. (1991). *The SMART retrieval system: Experiments in automatic document processing*. Prentice-Hall.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613–620.

Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. In *Proceedings of SIGIR'96*.

Smucker, M.D., & Allan, J. (2006). Find-similar: Similarity browsing as a search tool. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'2006)* (pp. 461–468).

Song, R., Liu, H., Wen, J. -R., & Ma, W. -Y. (2004). Learning block importance models for web pages. In *Proceeding of the thirteenth world wide web conference (WWW 2004)* (pp. 203–211).

Tombros, A., & Ali, Z. (2005). Factors affecting web page similarity. In *Proceedings of ECIR2005*.

van Rijsbergen, C. J. (1979). *Information retrieval*. London: Butterworths.

Xue, G. -R., Zeng, H. -J., Chen, Z., & Yu, Y. (2004). MRSSA: An iterative algorithm for similarity spreading over interrelated objects. In *Proceedings of CIKM2004*.

Yu, S., Cai, D., Wen, J. -R., Ma, W. -Y. (2003). Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of the twelfth international world wide web conference (WWW2003)*.

Zhang, B., Li, H., Liu, Y., Ji, L., Xi, W., Fan, W., Chen, Z., Ma, W. -Y. (2005). Improving web search results using affinity graph. In *Proceedings of SIGIR2005*.

Zhou, D., Weston, J., Gretton, A., Bousquet, O., Schölkopf, B. (2003). Ranking on data manifolds. In *Proceedings of NIPS-2003*.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., Schölkopf, B. (2003). Learning with local and global consistency. In *Proceedings of NIPS-2003*.