SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2014, Vol. 57 052107:1–052107:17
doi: 10.1007/s11432-012-4781-6

XDist: an effective XML keyword search system with re-ranking model based on keyword distribution

GAO Ning[1,2], DENG ZhiHong[1]* & LÜ ShengLong[1]

[1]*Key Laboratory of Machine Perception (Ministry of Education), School of Electronics Engineering and Computer Science, Peking University, Beijing* 100871*, China;*
[2]*College of Information Studies, University of Maryland, College Park, MD* 20742*, USA*

Received December 17, 2013; accepted February 24, 2014

Abstract Keyword search enables web users to easily access XML data without understanding the complex data schemas. However, the native ambiguity of keyword search makes it arduous to select qualified relevant results matching keywords. To solve this problem, researchers have made much effort on establishing ranking models distinguishing relevant and irrelevant passages, such as the highly cited TF*IDF and BM25. However, these statistic based ranking methods mostly consider term frequency, inverse document frequency and length as ranking factors, ignoring the distribution and connection information between different keywords. Hence, these widely used ranking methods are powerless on recognizing irrelevant results when they are with high term frequency, indicating a performance limitation. In this paper, a new searching system XDist is accordingly proposed to attack the problems aforementioned. In XDist, we firstly use the semantic query model maximal lowest common ancestor (MAXLCA) to recognize the returned results of a given query, and then these candidate results are ranked by BM25. Especially, XDist re-ranks the top several results by a combined distribution measurement (CDM) which considers four measure criterions: term proximity, intersection of keyword classes, degree of integration among keywords and quantity variance of keywords. The weights of the four measures in CDM are trained by a listwise learning to optimize method. The experimental results on the evaluation platform of INEX show that the re-ranking method CDM can effectively improve the performance of the baseline BM25 by 22% under iP[0.01] and 18% under MAiP. Also the semantic model MAXLCA and the search engine XDist perform the best in their respective related fields.

Keywords XML, keywords search, information retrieval, ranking model, keyword distribution, evaluation

Citation Gao N, Deng Z H, Lü S L. XDist: an effective XML keyword search system with re-ranking model based on keyword distribution. Sci China Inf Sci, 2014, 57: 052107(17), doi: 10.1007/s11432-012-4781-6

1 Introduction

Keyword search has been proved to be a user-friendly way of querying XML documents, since it allows users to pose queries without knowing the complex structure schema and query language, such as XQuery [1] and XPath [2], for example. In particular, the effect of the ranking module in terms of result relevance is one of the most crucial parts in XML keyword search engine. A global evaluation platform, initiative for the evaluation of XML retrieval (INEX)[1)], has been launched since 2002 for researchers from information retrieval, database and other relative research fields to compare the effect of their XML ranking engines.

*Corresponding author (email: zhdeng@cis.pku.edu.cn)

1) http://www.inex.otago.ac.nz/.

<section>
Japanese art covers a wide range of art styles and media, including ancient pottery, sculpture in wood and bronze, ink painting on silk and paper, and a myriad of other types of works of art. Historically, Japan has been subject to sudden invasions of new and alien ideas followed by long periods of minimal contact with the outside world. Over time Japanese developed the ability to absorb, imitate, and finally assimilate those elements of foreign culture that complemented their aesthetic preferences. The earliest complex art in Japan was produced in the 7th and 8th centuries A.D. in connection with Buddhism. In the 9th century, as the Japanese began to turn away from China and develop indigenous forms of expression, the secular arts became increasingly important; until the late 15th century, both religious and secular arts flourished.
Painting is the preferred artistic expression in Japan, practiced by amateurs and professional alike. Until modern times, the Japanese wrote with a brush rather than a pen, and their familiarity with brush techniques has made them particularly sensitive to the values and aesthetics of painting. With the rise of popular culture in the Edo period, a style of woodblock prints called ukiyo-e became a major art form and its techniques were fine tuned to produce colorful prints of everything from daily news to schoolbooks. The Japanese, in this period, found sculpture a much less sympathetic medium for artistic expression; most Japanese sculpture is associated with religion, and the medium's use declined with the lessening importance of traditional Buddhism.
</section>

Figure 1 (Color online) Relevant passage.

In an XML keyword search engine, the performance of the ranking module directly resolves the user experience, and thus it is one of the most crucial factors that determine the effect of the search engine. Many research efforts have been conducted in ranking methods of XML keywords search, such as TF*IDF [3] and BM25 [4]. Most of these ranking functions focus on extracting the statistic factors from the results, such as term frequency, inverse document frequency and length. Under the consideration of these ranking methods, the more keywords matches appear, the more relevant a result is.

However, the native ambiguity problem of keywords disarranges this procedure of ranking. In detail, XML keyword queries usually have ambiguities in interpreting the search intention and recognizing relevant documents. It is because of two reasons: (1) a keyword appearing in query and candidate documents can present different meanings; (2) a keyword appearing in one document can carry different meanings. Figure 2 shows a typical example. Query keywords Japanese and art have been highlighted. The search intention of the query is about the traditional or modern art of Japanese. However, the keyword Japanese in the parenthesis in line 12 indicates the nationality of Toshihiro Kawamoto. Additionally, under tag <references>, the keywords Japanese presents that those reference websites are in Japanese language, whose meanings are totally different from the keyword's in query. If simply treating these keywords as the same, this irrelevant passage will be falsely recognized as relevant.

For a further analysis, given a query Japanese art, the relevant passage in Figure 1 concentrates on the history of Japanese art and specifically, introduces the painting art form. On the contrary, the irrelevant passage in Figure 2 mainly depicts an artist called Mark Giambruno who is interested in Japanese anime. Because the lengths and total term frequencies of these two passages are similar, the existing ranking methods emphasizing term frequency, such as TF*IDF and BM25, would be arduous to separate these passages. Hence, under the aforementioned conditions, the traditional ranking methods based on the statistic of term frequency might not be able to correctly separate relevant and irrelevant passages.

To solve the aforementioned problem, some researchers claim that phrase and term proximity can effectively improve the top precision [5–7]. Peng et al. proposed a statistical language model [8], where the keywords matches in the results are considered to be valid only if the matches appear in the same order as in the query. Song et al. [9] evaluated the relevance of a result by splitting it into flexible spans of terms and subsequently incorporating the weights, based on the term frequency in the span and length, into the BM25 function. Svore et al. [10] proposed a research on exploring the improvement of phrase and term proximity by introducing novel ranking features based on flexible proximity terms with machine learning ranking models, following the work of [9]. Rasolofo et al. proposed a method [11] using proximity measurement combined with BM25, too. However, in [11], the term proximity weight is only computed between two keyword matches satisfying two conditions: (1) their distance is lower than or equal to 5; (2) they appear in the same order as in the submitted query.

In this paper, to escape the limitation of traditional ranking methods and further improve the effect of search engine, we observe the data collection and discover that the distribution of the keyword matches in results plays a crucial role in picturing the theme of the passage. Moreover, four detailed statistical characteristics based on distribution are generated to embrace advantageous capability on distinguishing

<section>
Born in Placerville, CA, the American animator artist Mark Giambruno grew up in Sacramento, CA before moving to the San Francisco Bay area for a number of years. In college, he majored in art and electronics, and began a career in computer graphics in 1990. After a few years doning presentation and web graphics, he transitioned into video games, and worked as artist, writer, art director, project manager or consultant on 21 different game titles, including The Incredibles: When Danger Calls, Mechwarrior 3, Zork: Nemesis and The Daedalus Encounter.
At the same time, he became involved in a number of writing projects, including the monthly Animata column for InterActivity magazine and books on 3D graphics. In the mid-nineties, he was attracted to the world of Japanese anime and manga, and amassed a large collection of Japanese books and DVDS. He even studied the language, and eventually started to word on adaptations of Japanese novels, manga and anime. While still at Mondo, he worked with anime legend Toshihiro Kawamoto (Japanese) of Cowboy Bebop fame on some character designs for an unreleased internet minishow.
</section>
<references>
References
"Kanon's visual novel official website" (in Japanese). Key. Retrieved on 2007-11-30.
"Air's visual novel official website" (in Japanese). Key. Retrieved on 2007-11-30.
"Clannad's visual novel official website" (in Japanese). Key. Retrieved on 2007-11-30.
"Planetarian's visual novel official website" (in Japanese). Key. Retrieved on 2007-11-30.
"Kanon's visual novel official website" (in Japanese). Key. Retrieved on 2007-11-30.
"Tomoyo After's visual novel official website" (in Japanese). Key. Retrieved on 2007-11-30.
"Little Busters! products page" (in Japanese). Key. Retrieved on 2007-11-30.
</references>

Figure 2 (Color online) Irrelevant passage.

relevant and irrelevant passages. Therefore, based on the distribution of the keywords in results, a new re-ranking model combined distribution measurement (CDM) is proposed, in which four criterions are used to measure the relevance of an element:

• **Criterion 1.** Term proximity (TP). The number of sentences that contain different keywords is counted. A passage with more sentences that contains all keywords will be more relevant.

• **Criterion 2.** Intersection of keyword classes (IKC). The matches in passage of a certain keyword are firstly categorized into several subsets. Then the intersections of these subsets are calculated. The more characters contained by the intersection sets, the more relevant a passage is.

• **Criterion 3.** Degree of integration among keywords (DIK). These passages with high degree of integration will be more concentrating on one certain theme and should be given higher priority in the returned list.

• **Criterion 4.** Quantity variance of keywords (QVK). These passages, whose numbers of different keywords varying significantly, should be penalized.

Moreover, we will propose a new effective XML keyword search engine XDist. In XDist, we firstly introduce the semantic model MAXimal lowest common ancestor (MAXLCA) [12] to define the candidate results of a given query. In [12], MAXLCA outperforms in the experimental comparison with the baselines: XRANK, SLCA and XSeek. Then these results are ranked by BM25. Next, the top several results in the returned list are re-ranked by the distribution based measurement CDM, in which four criterions are taken into consideration. The weights of these four measures in the re-ranking method CDM are trained by a listwise learning to optimize methods we proposed [13], which will be simply introduced in Subsection 3.4.

The primary contributions of this paper include: (1) a re-ranking method CDM is proposed that utilizes the distribution of keywords in candidate result to measure the relevance between the passage theme and search intention, (2) four relevance measures are observed and concluded, which are confirmed to be advantageous in improving the performance by the experiments, (3) the effect of the approach on the evaluation platform of INEX is verified.

The rest of this paper is organized as follows. In Section 2, we introduce the related works. Section 3 gives the preliminary knowledge. The four relevance measures are discussed in detail in Section 4. In Section 5 we report our experimental results. Section 6 contains the conclusion and future work.

2 Related work

In recent years, several search engines have been proposed by researchers to retrieve the XML collections. Here we introduce several highly cited systems. XRANK was presented by Guo et al. [14]. XRANK

extends the PageRank hyperlink metric to XML ranking. Apparently, PageRank does not deal with the ambiguity problem. In XRANK, a node in XML tree is designated as a result node only if it contains at least one occurrence of each keyword in its subtree, after excluding the nodes in its descendants that already contain all the keywords. The formal definition is as follow:

Definition 1. Given a keyword query $Q = \{k^1, \ldots, k^q\}$, an XML tree Xtree, we assume that $V^i(1 <= i <= q)$ is the set of nodes that directly contains keyword k^i in Xtree. LCA(Q, Xtree) is defined as LCA$(Q, \text{Xtree}) = \{n | \exists(v_1 \in V^1, \ldots, v_q \in V^q), n$ is the lowest common ancestors of $\{v_1, \ldots, v_q\}$.

Definition 2. Given a keyword query $Q = \{k^1, \ldots, k^q\}$, an XML tree Xtree, XRANK(Q, Xtree) is defined as follows: XRANK$(Q, \text{Xtree}) = n | n^* \in \text{LCA(Q, Xtree)}$, where $n*$ is node n after excluding all its descendant nodes which belong to LCA node set.

XKSearch was put forward by Xu et al. [15]. XKSearch defines SLCA as query semantic model. For a query, a node in XML tree is considered as a result node in SLCA only if it contains at least one occurrence of each keyword in its subtree, and none of its descendants does. However, XKSearch does not address the ranking problem. The formal definition of SLCA is as follow:

Definition 3. Given a keyword query $Q = \{k^1, \ldots, k^q\}$, an XML tree Xtree, SLCA(Q, Xtree) is defined as SLCA$(Q, \text{Xtree}) = \{n | n \in \text{LCA}(Q, \text{Xtree}) \cap \neg(\exists n' \in \text{LCA}(Q, \text{Xtree}), n \succ n')\}$, where $n \succ n'$ means n is an ancestor of n'.

XSeek was introduced by Liu et al. [16]. In XSeek, nodes in XML tree are grouped into three categories: entity node, attribute node and connection node.

Definition 4. Entity node: If a node has siblings under the same name, then this indicates a many-to-one relationship with its parent node, and is considered to represent an entity. E.g. <workshop>, <paper> in Figure 4 are considered as entity nodes.

Definition 5. Attribute node: If a node does not have siblings of the same name, and it has one child, which is a value, then it is considered to represent an attribute. E.g. <date>, <title>, <editors>, <author> are defined as attribute nodes.

Definition 6. Connection node: A node is a connection node if it represents neither an entity nor an attribute. E.g. <proceedings> in Figure 4 is a connection node.

Given a keyword query, XSeek scans the candidate result list, and then replaces each non-entity node with the nearest entity node on its path to the root. This process guarantees that each node returned to the user is an entity node. Same as XKSearch, the ranking strategy is not discussed in XSeek.

XReal was put forward by Bao et al. [17]. XReal utilizes the statistics of underlying XML data to attack the ranking problem. Firstly it identifies the search for nodes and search via nodes of a query, and then the search engine ranks the individual matches of all candidate results by using an XML TF*IDF strategy. Nevertheless, XReal is unable to solve the ambiguity problem.

3 Preliminaries

3.1 System framework

Figure 3 describes the framework of our search engine. The inverted index of the data collection is initially processed in the background. Afterwards, when user submits a query to the interface, the search engine firstly retrieves the relevant elements according to the definition of the semantic model maximal lowest common ancestor (MAXLCA). The extracted elements are disordered results of the query. Thus, we use a ranking model BM25 to rank these disordered element results, and the output of the processing is a ranked list. To further improve the effect of the ranking module, we re-rank the top several results in the ranked list by distribution measurements. In distribution measurements, there are four criterions based on the distribution of keywords taken into consideration, explicitly introduced in Section 4, and the weights of these four measurements in the final ranking function are trained by a learning to optimize

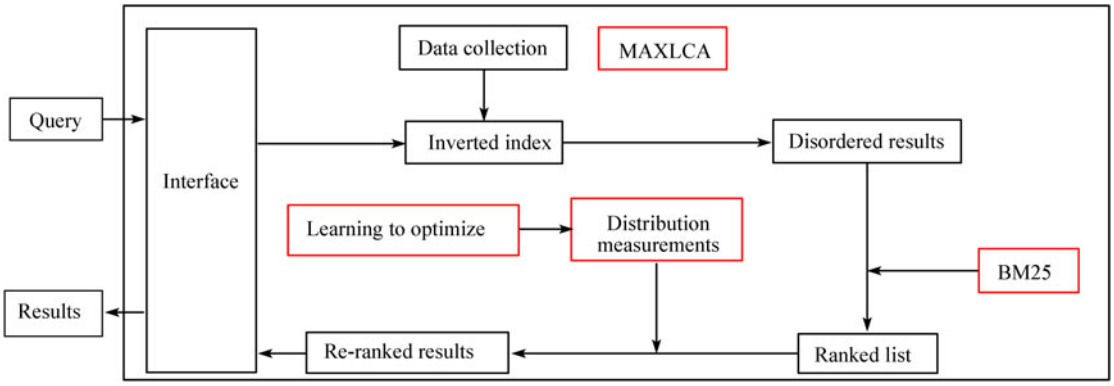

Figure 3 (Color online) System framework.

method. Finally, the re-ranked results are returned to the user as searching results.

3.2 Maximal lowest common ancestor (MAXLCA)

In this section, we will simply introduce the semantic model used in XDist. Different from the traditional HTML retrieval, the returned results of XML retrieval can be elements. Semantic query models are used to define the returned element results of a given query. Several approaches have been proposed to identify relevant results, such as XRANK, SLCA, XSeek and so on.

In this paper, we use MAXLCA [12] as the semantic query model. In [12], MAXLCA outperforms in the experimental comparison with XRANK, SLCA and XSeek. To sum up, in MAXLCA, only the maximal LCA element of a document is returned. The formal definitions are as follow:

Definition 7. Given a keyword query $Q = \{k^1, \ldots, k^q\}$, an XML tree Xtree, we define MAXLCA(Q, Xtree) as $\text{MAXLCA}(Q, \text{Xtree}) = \{n | n \in \text{LCA}(Q, \text{Xtree}) \cap \neg(\exists n' \in \text{LCA}(Q, \text{Xtree}), n \succ n')\}$, where $n \succ n'$ means n is an ancestor of n'.

The definition of MAXLCA maximal preserves the relevant information in a passage and minimizes the irrelevant information. Further, since for each XML tree, there is only one MAXLCA node recognized, the overlap problem is avoided. Figure 4 is a sample XML tree marked with Dewey ID [14]. Given a query Albert Einstein, the nodes 0.3.1.0, 0.3.1, 0.3.2.0.0, 0.3.2.0, and 0.3 are LCA nodes. The node 0.3.1.0 and 0.3.2.0.0 are the returned results according to the definition of XRANK, SLCA and XSeek. On the other hand, the only MAXLCA node in this XML tree is 0.3. Hence, the node 0.3 is extracted from the relevant passage and returned as a retrieval result. The algorithm of finding MAXLCA is proposed in [12].

3.3 Basic ranking function BM25

In information retrieval, BM25 is a highly cited ranking function used by search engines to rank documents according to their relevance to a given search query. Though BM25 is originally proposed to rank the HTML format documents, it was introduced to XML documents ranking field in recent years. In the last three years of INEX Ad Hoc track, all the search engines that perform the best [18–20] use BM25 as basic ranking function. Considering its outstanding ranking effect, it is better to fit BM25 into our new ranking model rather than substitute it. In this paper XDist takes BM25 as a first-step ranking tool. The candidate results are initially ranked by BM25. Afterwards, the top several returned results are re-sorted by a re-ranking function combined with four distribution measurements. The formal definition of BM25 is presented as follow

$$\text{ps}(\text{e}, Q) = \sum_{t \in Q} W_t \cdot \frac{(k+1) \cdot tf(t, \text{e})}{k \cdot (1 - b + b \cdot \frac{\text{len(e)}}{\text{avel}}) + tf(t, \text{e})}. \tag{1}$$

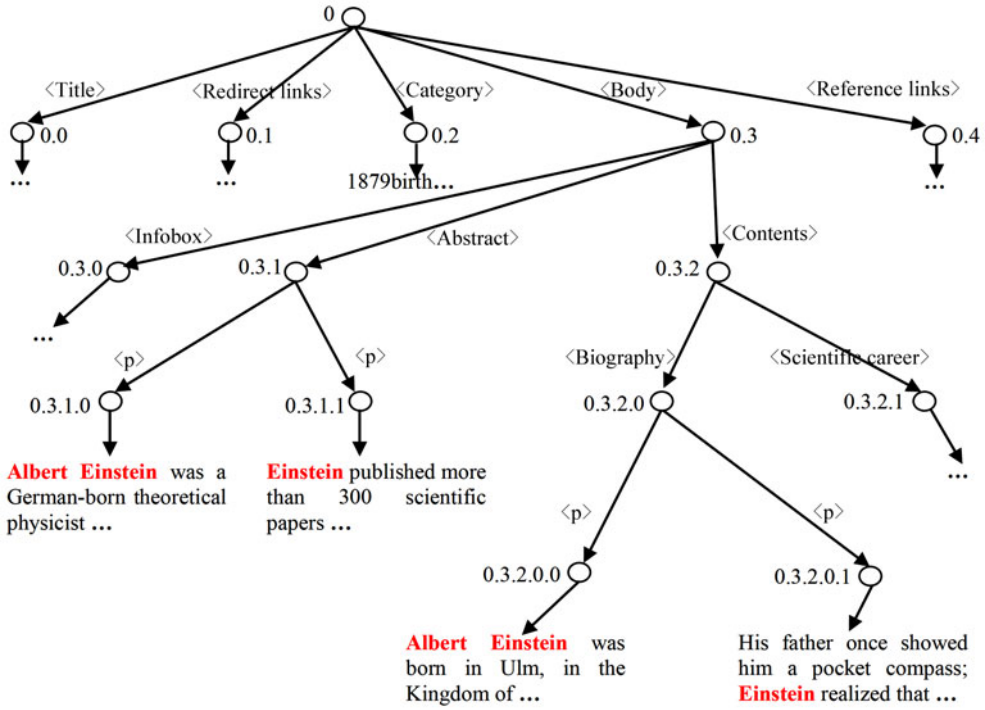

Figure 4 (Color online) Sample XML tree.

$$W_t = \log \frac{Nd}{n(t)}. \tag{2}$$

In the formula, $tf(t, \mathrm{e})$ is the frequency of keyword t appearing in element e; Nd is the number of files in the collection; $n(t)$ is the number of files that contain keyword t; len(e) is the length of element e; avel is average length of elements in the collection; Q is a set of keywords; ps(e,Q) is the predicted relevance score of element e corresponding to query Q. k and b are two free parameters to be tuned according to the data set.

3.4 Learning to optimize

As defined in formula (3), the re-ranking method combined distribution measurement (CDM) used in the our search engine XDist combines four features linearly and shows remarkable effect on distinguishing relevant and irrelevant passages. In (3), CDM(e, Q) is the re-ranking score of element e corresponding to query Q; TP considers the term proximity; IKC considers the intersection of keyword class; DIK considers the degree of integration among keywords and QVK is the quantity variance of keywords. In addition, α, β, γ and δ denote the weights of each measures in the re-ranking function, trained respectively by the listwise optimizing method [13]. Since all the experiments are processed on the evaluation platform of INEX, the learning procedure takes English Wikipedia documents as experimental data collection. 115 topics in the query set of INEX 2009 are taken as training set, and the optimized weights are used in the testing of INEX 2010 queries.

$$\mathrm{CDM}(\mathrm{e}, Q) = \alpha \cdot \mathrm{TP} + \beta \cdot \mathrm{IKC} + \gamma \cdot \mathrm{KIM} + \delta \cdot \mathrm{QVK}. \tag{3}$$

As a compendium, the process of optimization is described as follow: In training, there is a set of query $Q = \{q^1, q^2, \ldots, q^m\}$. Each query q^i is associated with a list of candidate elements (top several results returned by BM25 in this paper). $E_i = (\mathrm{e}^i_1, \mathrm{e}^i_2, \ldots, \mathrm{e}^i_{n(i)})$, where e^i_j denotes the j-th candidate result to query q^i and $n(i)$ is the size of E_i. Moreover, each candidate elements list E_i is associated with a ground-truth list $G_i = (g^i_1, g^i_2, \ldots, g^i_{n(i)})$, indicating the relevance score of each elements in E_i.

Given that the data collection we used only contains information of whether or not the passages in a document are relevant, we apply the F-measure [21] to evaluate the ground truth score. Furthermore,

for each query q^i, we use the re-ranking function to get the predicted relevant score of the elements, recorded in $R_i = (r_1^i, r_2^i, \ldots, r_{n(i)}^i)$. Then each ground-truth score list G_i and predicted score list R_i form an instance. The loss function is defined as the euclidean distance among standard results lists D_i and search results lists R_i, presented as formula (4) and (5).

$$D_i = \sum_{i=1}^{m} L(G^i, R^i), \tag{4}$$

$$L(G^i, R^i) = \sqrt{\sum_{j=1}^{n}(i)(r_j^i - g_j^i)^2}. \tag{5}$$

In each training epoch, re-ranking function is used to compute the predicted score R_i. Then the learning module replaces the current weights with the new weights tuned based on the loss between G_i and R_i. Take α as an example. The tuning step is defined in the following formula

$$\alpha \leftarrow \alpha - \varphi \cdot \Delta\alpha, \tag{6}$$

$$\Delta\alpha = \sum_{q=1}^{m} \frac{\partial L(G^i - R^i)}{\partial \alpha} = \sum_{q=1}^{m} \frac{\sum_{j=1}^{n(i)}(r_j^q - g_j^q) \cdot \mathrm{DK}}{\sum_{j=1}^{n(i)}(r_j^q - g_j^q)^2}.$$

Here φ is set to control the learning speed. The iteration stops when either the limit cycle index is reached or the parameters do not change any more. The learning procedure outputs the optimized weights of each measure. Detailed explanations and definitions can be found in [13].

4 Four relevance measurements

In this section, we will introduce the four relevance measurements based on the distribution of keywords in results. All analysis and examples are based on the relevant passage in Figure 1, irrelevant passage in Figure 2 and a given query Japanese art.

4.1 Term proximity (TP)

Intuitively, the keywords submitted to the search engine are usually extracted from a question sentence. For example, a user who wants to find something about the Japanese art will probably submit a query Japanese art. Similarly, president United States will be issued if a user needs information about the president of the United States for a history assignment. Inasmuch that the keywords in query generally stem from a phrase or a question user asked, it can be conjectured that one passage containing this phrase should be more relevant to the query topic than one passage that does not. For example, the subject of the first sentence in Figure 1 is the phrase Japanese art. However, there is no Japanese-art pair appearing in one sentence in Figure 2.

$$\mathrm{TP} = \sum_{(t^i, t^j) \in P} W_{(t_i, t_j)} \cdot \frac{(k+1) \cdot tf(t^i, t^j)}{k \cdot (1 - b + b \cdot \frac{\mathrm{len(e)}}{\mathrm{avel}}) + tf(t^i, t^j)}, \tag{7}$$

$$W_{(t^i, t^j)} = \log \frac{1}{2^{|j-i|}}. \tag{8}$$

Inspired by this and the widely used BM25, we define the term proximity measure in formula (7). Given a query $q = \{t_1, \ldots, t_q\}$, TP is the relevance between a candidate result r and q. P is the set of all query term pairs. The weight of each term pair is supposed to follow an autoregressive pattern [22], defined in (8). It is based on a simple assumption: the closer t^i and t^j are in the query, the higher their weight is $tf(t^i, t^j)$ is the number of sentences that contain t^i and t^j. Thusly, the measurement TP is designed to prefer outcomes with adjacent keywords appearing in one sentence.

Table 1 Extended inverted index example

Term	Document ID	Paragraph	Sentence	Frequency
Japanese	1	1	1	1(1)
			3	1(63)
			5	1(108)
Art	1	1	1	3(2,8,35)
			4	1(86)
				2(123,136)

To accelerate the response speed and improve the efficiency, the search engine does not need to scan each sentence when measuring the distance among keywords. The solution is to specialize the location of terms recorded in inverted index to paragraphs and sentences. Further the appearance locations of each keyword are recorded with the frequency.

Table 1 is an explanation example for the extended inverted index, indicating the locations of keywords Japanese and art in the first paragraph of the relevant passage. The numbers in the brackets present the position of the corresponding keyword. As a result, whenever the location information of keywords is needed, the position vector can be extracted from the extended inverted index directly, without much extra time exhausting. For example, in the measurement of DK, the intersection of the keywords' sentence lists is the sentences that contain the keywords at the same time.

4.2 Intersection of keyword classes (IKC)

Although the keywords in the query are irrelevant to each other from the perspective of the dictionary meanings, the intersection of these keywords' expression range reveals the search intention. Here we assume that the keywords' expression range is marked by their appearances. For example, in Figure 1, the two keywords Japanese and art appear in both the two paragraphs, indicating that their expression ranges are completely integrated. On the other hand, in Figure 2, the keyword art appears in paragraph 1 and Japanese appears in paragraph 2 as well as the <references>. Thus there is significant difference between the expression ranges of art and Japanese. Here under the measurement of IKC, we assume that (1) if there is no intersection between different keywords classes, the keywords are used to express different topics that the passage is irrelevant; (2) if the classes of different keywords completely coincide, the passage is relevant with high probability. Therefore, IKC measures the intersection of different keyword classes. The more characters contained by the intersection sets, the more relevant a pas-sage is.

To simulate the phenomenon mentioned, we firstly divide the keywords into classes. Two appearances of a same query keyword within a distance of 30 words are grouped into one class. For example, in Figure 1, the distance between the first and the second art is 6 so that these two matches are defined in the same art class. In this way, the keywords could be categorized into several subsets, each of which is considered to be concentrating on one topic.

Given a query $q = \{k^1, \ldots, k^q\}$, the matches locations of each keywords can be directly extracted from the extended inverted index introduced in Table 1. The locations are recorded in corresponding vectors for each keyword: $\{V^1, \ldots, V^q\} = \{V_1^1, \ldots, V_{n_1}^1, \ldots, V_1^q, \ldots, V_{n_q}^q\}$, in which n_i is the size of vector V^i and V_j^m is the j-th appearance of the m-th keyword. For each keyword, indexed as m for instance, we firstly cluster its matches discussed above into several sets according to its corresponding position information V^m, the clusters thus are recorded as: $S^m = \{S_1^m, \ldots, S_{\mathrm{nc}_m}^m\}$, where nc_m is the number of the subsets categorized for keyword m. S_i^m is the i-th cluster of $V^m = \{V_1^m, \ldots, V_{n_m}^m\}$. Further, we define the information set of the clustered sets of $\{V^1, \ldots, V^q\}$ as

$$\mathrm{SV}^m = \mathrm{interval}(S^m) = (\mathrm{SV}_1^m, \ldots, \mathrm{SV}_{\mathrm{nc}_m}^m),$$

where $\mathrm{SV}_i^m = (\min(S_i^m), \max(S_i^m))$ is the minimal open interval covering S_i^m. $\mathrm{SV} = \{\mathrm{SV}^1, \ldots, \mathrm{SV}^q\} = \{\mathrm{SV}_1^1, \ldots, \mathrm{SV}_{nc_1}^1, \ldots, \mathrm{SV}_1^q, \ldots, \mathrm{SV}_{nc_q}^q\}$. SV is treated as the information set of $\{S^1, \ldots, S^q\}$, sharing the most basic information of S.

Then we could define the intersection of multiple different sets as

$$I = \mathrm{SV}^1 \bigcap ... \bigcap \mathrm{SV}^n = I_1 \bigcup ... \bigcup I_{n_I}, \tag{9}$$

I is the intersection of several open intervals. $\{I_1, \ldots, I_{n_I}\}$ have two properties:

Property 1. Each of $\{I_i, i = 1, \ldots, n_I\}$ is an open interval: $I_i = (a_i, b_i)$.

Property 2. $a_1 < b_1 < a_2 < b_2 < \ldots < a_{n_I} < b_{n_I}$.

The weight of each intersection set I_i could also be decided according to the number of different keyword sets it contains, denoted by t_i. The more it contains, the more important it is. In mathematical way,

$$t_i = \sharp\{j : S_j \cap I_i \neq \emptyset\},$$

where $\sharp\{.\}$ indicates the number of elements in a set. In this way, the IKC is defined as

$$\mathrm{IKC} = \sum_{i=1}^{n_I} \mathrm{e}^{t_i} \cdot W_i, \tag{10}$$

where n_I is the total number of intersection sets, $W_i = b_i - a_i$ denotes the size of I_i, and t_i discussed as above represents I_i's weight. Same as formula (7), base e is set to 2.718281828.

4.3 Degree of integration among keywords (DIK)

The topic of Figure 1 is the art of Japanese, so that the matches of the two keywords Japanese and art should appear alternatively to describe the theme. However in Figure 2, the passage is about an artist Mark Giambruno who is interested in Japanese anime. The keyword matches of art in paragraph 1 are used to present the identity of Mark as an artist. The appearances of Japanese in paragraph 2 and under <references> are used to describe the anime and the accordingly websites. There is no semantic intersection between art and Japanese, so that there is also no intersection between the appearances of these two keywords' sets since they focus on different topics.

It is reasonably supposed that the passage with high degree of integration will be more concentrating on one certain topic. On contrary, the passage without keywords' intersection will probably present several discrete subjects irrelevant to the query. Considering the most extreme circumstances, there are two documents. Different keywords in the first document appear alternatively, while in the second document the different keywords appear in different parts of the documents. It is reasonable to assume that the topic of the first document matches the query better. Thus, the re-ranking method should make sure that the passages satisfying degree of integration have high priority in the returning list. Inspired by that, we define the third distribution standard as the degree of integration among keywords.

A statistic method, Wilcoxon rank sum test, also called the Mann-Whitney U test [23,24], is applied to measure the integration of the matches of keywords. Wilcoxon rank sum test is a non-parametric statistical hypothesis test initially used for assessing whether two independent samples of observations have equally large values. In measuring the degree of keywords' integration, we firstly extract the positions of keywords' matches in the results and then preserve them in the vectors respectively. Given two vectors, each of them recording the appearing positions of a keyword, we use Wilcoxon rank sum test to measure the integration of the two keywords and examine whether these two groups of matches appear alternatively.

According to the definition of Wilcoxon rank sum test, given a query $q = \{k_1, \ldots, k_q\}$ and the corresponding vectors of keywords $\{V^1, V^2, \ldots, V^q, q \geqslant 2\}$, whose positions in a document has been recorded as $\{V_1^1, \ldots, V_{n_1}^1, \ldots, V_1^q, \ldots, V_{n_q}^q\}$, where V^1 appears n_1 times and V^m appears n_m times, it is possible to detect the differences of distributions or equivalently testing whether the keywords are highly integrated in this document.

In detail, firstly, for any two keywords, $\{V^1, V^2\}$ for instance without loss of generality, the corresponding positions $V = \{V^1, V^2\} = \{V_1^1, \ldots, V_{n_1}^1, V_1^2, \ldots, V_{n_2}^2\}$ are ranked from low to high, denoted by $R = \{R^1, R^2\} = \{r_1, \ldots, r_{n_1+n_2}\}$. For example, if $V = \{V^1 = \{1, 63, 108\},\ V^2 = \{2, 8, 35, 86, 123, 136\}\}$,

then $R = \{1, 5, 7, 2, 3, 4, 6, 8\}$. Secondly, the Wilcoxon test statistic denoted by TA could be calculated as the sum of the ranks of the first vector, presented as

$$T_A = |R^1| = \sum_{i=1}^{n_i} r_{li}, \tag{11}$$

where $|.|$ represents the first norm of a vector and r_{li} is the rank of V_i^1 in R. According to the aforementioned example R, its corresponding $T_A = 1+5+7$, since its first element 1 appears first in R; the second element 63 appears in position 5 and the location of 108 in R is 7. Based on T_A, it is nicely proved [23] that if the null hypothesis, presented as

H_0: there is no difference for the distribution of the keywords' position V^1 and V^2, holds and the sample size $n = n_1 + n_2$ is large enough ($n \geqslant 10$), then a test statistic Z could be derived in formula (12)

$$Z = \frac{T_A - \frac{n_1 \cdot (n_1+n_2+1)}{2}}{\sqrt{\frac{n_1 \cdot n_2 \cdot (n_1+n_2+1)}{12}}}, \tag{12}$$

Z follows a standard normal distribution, with the property

$$P(Z \ll a) = \int_{-\infty}^{a} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{2}} \mathrm{d}x. \tag{13}$$

Then the probability that an event more extreme happens under H_0 is

$$\mathrm{Prob}_{1,2} = 1 - \int_{-|Z|}^{|Z|} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{2}} \mathrm{d}x, \tag{14}$$

$\mathrm{Prob}_{1,2}$ could be treated as the measure of integration of V^1 and V^2 in this document, and called $p-$value confined in statistics. The keywords are believed to be closer to each other if p-value is higher. Further, for q higher than 2, we could use the idea of multiple test adjustment with various related works such as Bonferroni and false discovery rate [25,26]. For simplicity, we use the idea of Bonforroni [25] methods here: for each two keywords $\{V^{m_1}, V^{m_2}, m_1 \neq m_2\}$, Prob_{m_1,m_2} could be calculated as above. In this way, we could derive $\binom{q}{2}$ measures of integration: $\{\mathrm{Prob}_{1,1}, \mathrm{Prob}_{1,2}, \ldots, \mathrm{Prob}_{2,3}, \ldots, \mathrm{Prob}_{2,q}, \ldots, \mathrm{Prob}_{q-1,q}\}$.

Then the adjusted $p-$value or the measure of integration for the whole q keywords is

$$\mathrm{DIK} = \mathrm{Prob}_v = \binom{q}{2} \cdot \min\{\mathrm{Prob}_{1,1}, \ldots, \mathrm{Prob}_{1,2}, \ldots, \mathrm{Prob}_{2,3}, \ldots, \mathrm{Prob}_{2,q}, \ldots, \mathrm{Prob}_{q-1,q}\}, \tag{15}$$

Prob_v is the value to test the integration of keywords as we need, considered as DIK in the combined distribution function. Here we note that under some regularity conditions,

$$P(\mathrm{Prob}_v < 0.05 | H_0) = 0.05.$$

4.4 Quantity variance of keywords (QVK)

The existing ranking functions based on term frequency, such as TF*IDF and BM25, will give high priority to the passages with high appearance of keywords. Nevertheless, given the query Japanese art, a passage about Chinese art where art appears 100 times and there is only 1 appearance of Japanese (indicating an art website in Japanese language in reference list, for example), will be recognized as totally relevant by the aforementioned two ranking functions due to its high term frequency. Undoubtedly, the aforementioned kind of passages whose numbers of different keywords vary significantly should be heavily penalized. Under the criteria of QVK, we examine the returned results with high term frequency and remove the ones whose numbers of different keywords extremely vary.

The information entropy, also referred as Shannon entropy [27], is referenced to measure the dispersion of the frequency of keywords. Information entropy is initially proposed to measure the average of information content. Here when considering the quantity variance of the keywords, we treat each match of the keyword in the result as an information carrier. Thus for a given query $q = (k_1, \ldots, k_q)$ and a candidate

result, whose matches of keywords have been extracted into the series of vectors $\{V^1, V^2, \ldots, V^q\}$, the information carried by each keyword are preserved in the appearances of the keyword matches. Here we define two assumptions:

• **Assumption 1.** The more frequently a keyword appears in the result, the more information it presents.

• **Assumption 2.** If the quantities of different keywords in the result show an extreme discrepancy, it should be given lower priority in the returned list.

Based on these two assumptions and the original definition of information entropy, we define the measurement of quantity variance of keywords as

$$\mathrm{QVK} = -\sum_{i=1}^{q} \frac{n_i + 0.5}{q} \cdot \log(\frac{n_i + 0.5}{q}). \tag{16}$$

Here n_i is the size of vector V^i, indicating the times of appearances of the i-th keyword in the result. Since the numbers of keywords in each topic range from 1 to 11, to avoid the unbalance introduced by different topics in learn procedure, an additional parameter q, the keywords number in the query, is applied to the formula.

5 Experiments

In this section, the XML data set used in comparison experiments is introduced first. In Subsection 5.2, we will compare the performance of the used semantic query model MAXLCA in this system with the other models defined in systems of XKSearch, XRANK and XSeek. Then in Subsection 5.3 we compare the individual and combined effect of the four re-ranking criterions with the baseline BM25. In Subsection 5.4, we focus on testing the influence of the number of re-ranking results on the performance. Further in Subsection 5.5, we will compare the results of XDist with other aforementioned systems, including XRANK, XKSearch, XSeek and XReal. Finally in Subsection 5.6, we give the efficiency of our search engine.

5.1 Data collection

The data collection used in the experiments consists of 2 666 190 English XML files from Wikipedia, used by INEX Ad Hoc Track. The total size of these files is 50.7 GB. The query set consists of 107 different topics from the topic collection of Ad Hoc track. Each query in the evaluation system is bound to a standard set of highlighted relevant content, which is recognized manually by the participants of INEX. In the experiments, the training regards these highlighted relevant content as ground truth results.

5.2 Comparison of semantic query models

In this section, we firstly compare the performance of the semantic query model MAXLCA with the other three models, SLCA, XRANK and XSeek, defined in the systems of XKSearch, XRANK and XSeek respectively. To avoid the influence of different ranking method, we use BM25 as ranking strategy uniformly.

Figures 5 and 6 show the performance of different semantic query model under iP and MAiP criterions respectively. The x-axis and y-axis of Figure 5 is recall and the corresponding precision. The y-axis of Figure 6 is MAiP (the average iP), and the exact values are shown in x-axis. As can be seen, no matter under which assessing standard, MAXLCA performs best. The definitions of SLCA and XRANK suggest that these two models focus on finding core relevant parts and rejecting the less relevant and irrelevant parts, which show benefit when the searching dataset is formed by documents with higher average length, such as chapters of books. However, for the datasets with lower average length, such as web pages, the content of each document will be probably concentrating on one certain topic, and thus the work of finding the relevant parts completely is more important than rejecting the less relevant parts. On the other hand, the definitions of entity, attribute and connection node in XSeek strictly depend on the labels

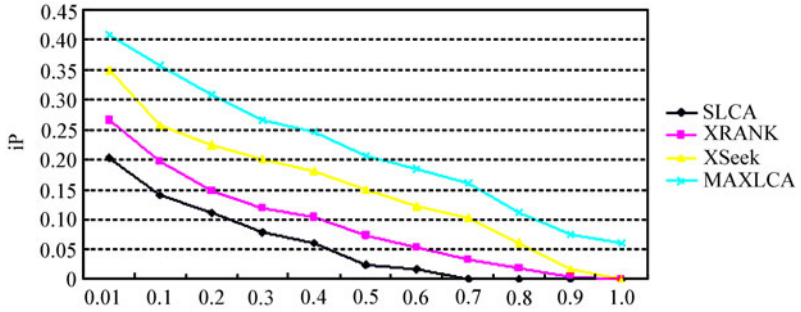

Figure 5 (Color online) Performance of semantic query models under iP.

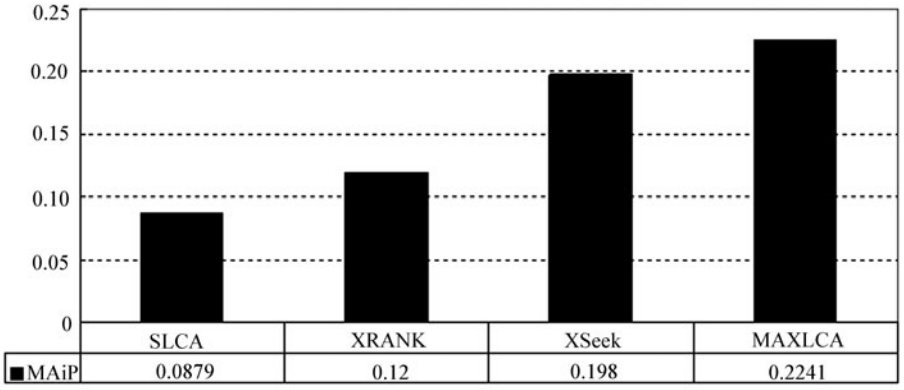

Figure 6 Performance of semantic query models under MAiP.

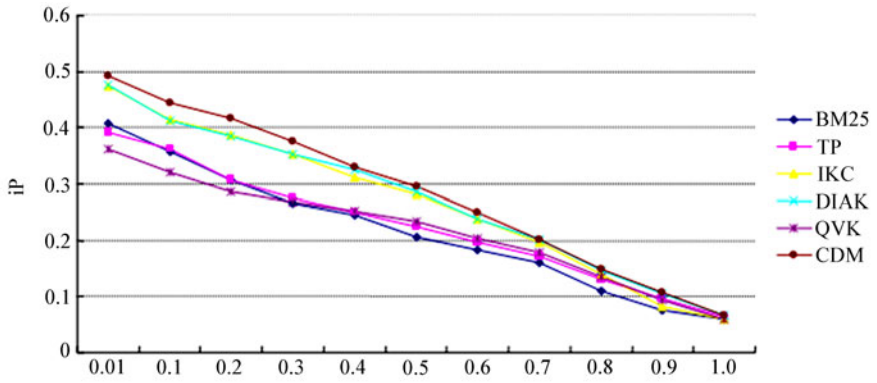

Figure 7 (Color online) Effect under iP.

of the XML documents. However, the labels in Wiki dataset are all structural tags, body, template and section for example, which are meaningless in distinguishing the nodes by their semantic differences. Therefore, MAXLCA avoids the disadvantages of the other comparing models, showing a better performance.

5.3 Performance of individual measurements

In this section, we investigate the individual performance of the four re-ranking measurements. As have been stated, the candidate results are initially ranked by the basic ranking function BM25, taken as the baseline of the comparison. Then the top 30 returned results are re-ranked according to the term proximity (TP), intersection of keyword classes (IKC), degree of integration among keywords (DIK) and quantity variance of keywords (QVK) respectively. Finally, CDM is the method using BM25 function as basic ranking strategy and the combined distribution function as re-ranking function, defined in formula (3). The experimental results under criterion iP and MAiP are illustrated in Figures 7 and 8.

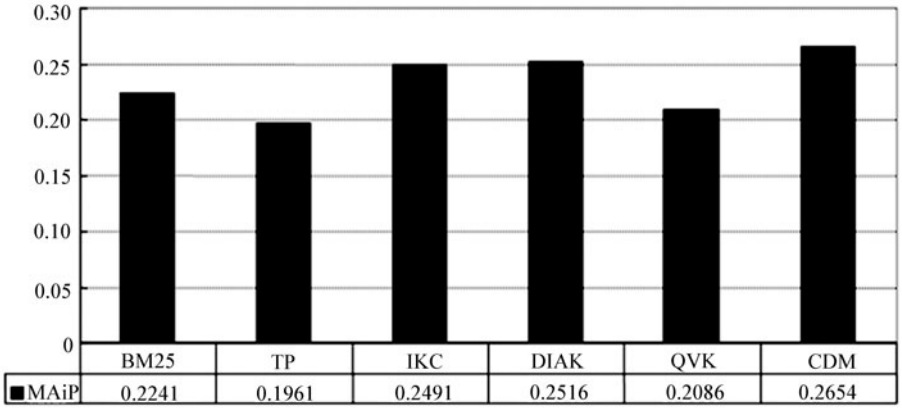

Figure 8 Effect under MAiP.

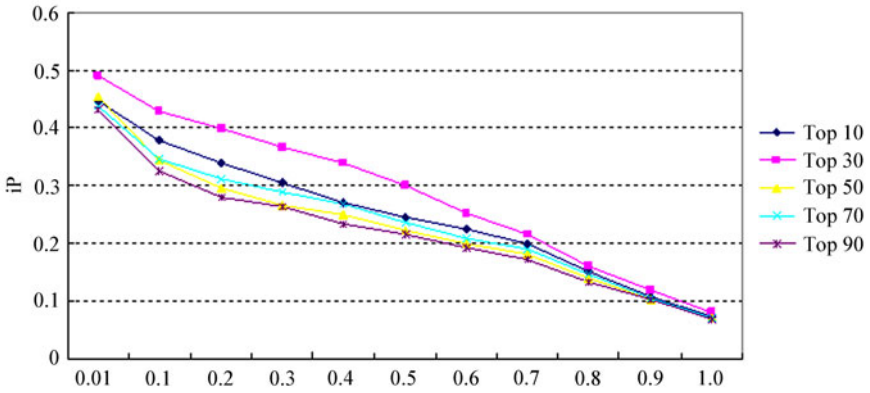

Figure 9 (Color online) iP under different re-ranking number.

It could be observed in Figures 7 and 8 that the effect of BM25 and the other distribution based measurements is: CDM > DIK > IKC > BM25 > TP > QVK. We initially expected that the individual effect of the four re-ranking measurement was much worse than BM25, since each of them only concentrates on one character of the passage and might be absent of the traditional proven effective features like term frequency and length. However, as confined in our experiments, the method considering intersection of the keyword classes and degree of integration among keywords perform even better than BM25 and there is no distinct disadvantage for the other two re-ranking methods. This phenomenon could be interpreted from the perspective that: for a given query, there would be a large quantity of candidate results returned. For example, the candidate results number of Japanese art is 32 872. Therefore, we can suppose that the top 30 results ranked by BM25 already outperform on the traditional properties such as term frequency and length. In this case, the additional consideration of distribution can improve the effect of separating relevant and irrelevant passages from a new perspective.

5.4 Re-ranking number

After testing the performance of individual distribution measurements and the combined distribution measurements with the basic BM25 ranking function, we have known that the combined ranking method CDM performs the best. In CDM, taking the returned list of BM25 as basic, the top several results should be re-ranked according to the definition of the combined distribution function in formula (3). However, the effect of CDM is partly determined by the number of re-ranking results. Thus in this section, we test the relationship of the performance of CDM and the re-ranking results number.

Figures 9 and 10 show the performance diversification of CDM with different re-ranking number under criterion iP and MAiP respectively. As shown, when the top 30 results are re-ranked according to the distribution function, the search engine performs the best. The idea of CDM is using BM25 as basic

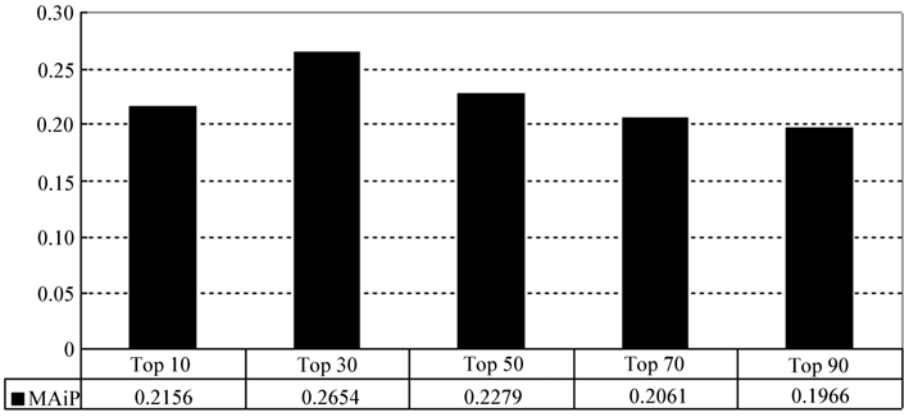

Figure 10 MAiP under different re-ranking number.

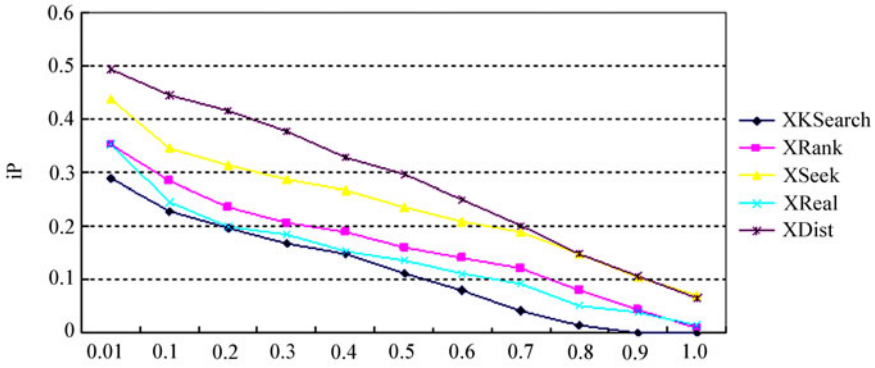

Figure 11 (Color online) Performance of systems under iP.

ranking method, and then re-ranking the top several returned results according to their distribution information. We should firstly admit that the factors used by BM25, such as term frequency, inverse document frequency and length, can effectively indicate the relevant results. The purpose of the re-ranking method is distinguishing the relevant and irrelevant documents when both of them have similar values on the factors used by traditional ranking methods. Thus rather than totally displace BM25, distribution measurements are more likely to be used to improve its performance. Then there will be the best re-ranking number, signed as Bn, leading to the most proper combination of BM25 and distribution measurements. If more than Bn results are re-ranked, then the ranking function will be biased in favor of the distribution measurements. Once more documents with low term frequency are re-ranked to the top of the list, the performance will decrease. On the other hand, if the re-ranking number is set smaller than Bn, the benefits of distribution measurements are not totally presented and then the performance will not improve much comparing with the basic ranking method BM25. Hence, there will be a most proper set Bn, keeping the balance of BM25 and distribution measurements. As the experiment shown, the set of Bn for the data collection used could be around 30.

5.5 Comparison with other systems

In this section, we compare our search engine XDist with the systems introduced in Section 2. As has been presented in Subsection 3.1, XDist takes the elements defined by MAXLCA as retrieval results. Afterwards, the disorder results are firstly ranked by BM25, and then re-ranked by the combined distribution measurements. Figures 11 and 12 present the iP and MAiP performance of the comparison systems. Since there are no discussing about the ranking module in the systems of XKSearch and XSeek, we implement BM25 as their ranking functions. As the experimental results show, the new ranking system XDist outperforms others.

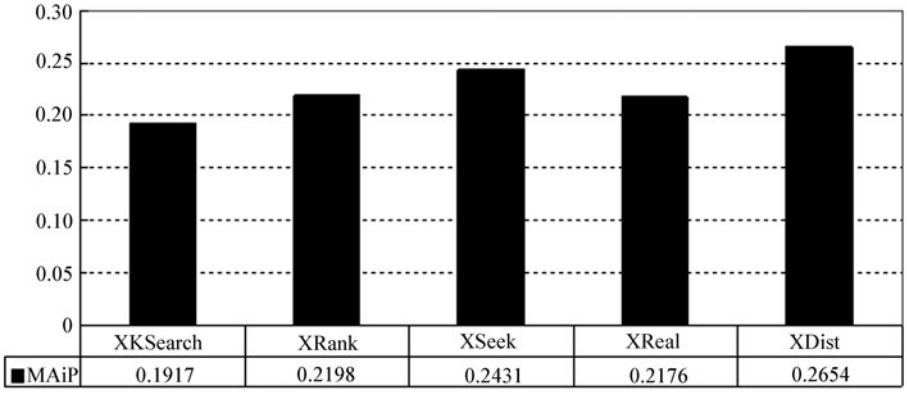

Figure 12 Performance of systems under MAiP.

Table 2 Different keywords number

Keywords number	1	2	3	4	5	6	7	8	9	10	11
Topics	2	40	30	23	9	0	1	0	1	0	1

Table 3 Efficiency of XDist (ms)

		Keywords number							
		1	2	3	4	5	7	9	11
Re-ranking number	0	415	576	601	498	392	189	98	32
	10	420	611	658	564	470	255	155	78
	30	426	647	715	631	566	300	206	132
	50	430	682	772	697	653	355	271	160
	70	437	718	830	764	740	411	329	203
	90	442	758	887	830	819	476	386	235

5.6 Efficiency

In this subsection, we discuss the efficiency of the search engine XDist. Compared with the original search engine using only BM25 as ranking method, the additional operations of extracting and computing distribution information will reduce the efficiency performance. Here we test the efficiency of XDist under different numbers of keywords. Table 2 shows the quantity of topics with different keywords numbers. As can be seen, there are in total 107 test queries, most of which have 2–4 keywords. The first row with numbers (1,2,. . .,11) indicates the number of keywords in the query. The first column with numbers (0,10,. . .,90) is the number of results to be re-ranked according to the definition of function (3). The value in each cell represents the time-consuming (ms) under particular keywords number and re-ranking number. Table 3 shows the efficiency of XDist under two dimensions: keywords number and the re-ranking documents number. The line signed as re-ranking number 0 is the engine using only BM25 as ranking method. Several regular patterns can be observed:

Pattern 1. With the increasing keywords, the time consumption of the engine using only BM25 as ranking function decreases.

The response time of a certain query is relevant with the number of returned results. The more relevant results are retrieved, the more time will be consumed in ranking module. With the increasing keywords, the relevant documents will apparently decreases. Thus with the additional of the keywords, the time consuming will be reduced.

Pattern 2. With the increasing re-ranking results number, the time consumption increases too, and there appears a linear relation between them.

The time consumption of computing the four distribution measurements is only relevant to the number of keywords matches in the results. Since there is no apparent decrease in the keywords matches when the

testing re-ranking number rises, the time consumption of the additional distribution computing operations increases linearly.

Pattern 3. Under the same re-ranking number set, with the increasing keywords number, the average addition of time consumption increases too.

According to the definitions of distribution measurements, the time consumed is highly relevant will the number of keywords. For example, when computing the integration of the keywords of a result, the time of executing Wilcoxon rank sum test is $\binom{P}{2}$, where P is the number of keywords in the query. Hence, the more keywords a query contains, the more time it costs in the re-ranking procedure.

6 Conclusion and future work

In this paper, we propose our high effect search engine XDist. In the system, we implement MAXLCA as the semantic query model. Then basic ranking function BM25 is used to rank the disordered results. Further, the top several results on the returned list are re-ranked by the combined distribution measurement (CDM), in which four criterions are considered: term proximity, intersection of keywords classes, degree of integration among keywords and quantity variance of the keywords. In detail, the weights of the four standards in the final re-ranking function are trained by a learning-to-optimize method. Moreover, we compare the effect of MAXLCA with other semantic models, the individual and combined effect of the four distribution measurements, the influence of re-ranking number and the system efficiency.

For the future work, there are numerous interesting research issues. Firstly, we will extend XDist to deal with other topics of keyword search over XML data, such as the recently proposed topics [28–35]. Secondly, we are interested in extending the idea of distribution measurements to the area of HTML documents searching.

Acknowledgements

This work was partially supported by National Natural Science Foundation of China (Grant No. 61170091) and National High Technology Research and Development Program of China (Grant No. 2009AA01Z136).

References

1 Chamberlin D, Florescu D, Robie J, et al. XQuery: a query language for XML. In: Proceedings of ACM SIGMOD, 2003. 682–682
2 W3C Recommendation. XML Path Language (XPath) Version 1.0, 1999
3 Carmel D, Maarek Y S, Mandelbrod M, et al. Searching XML documents via XML fragments. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, New York, 2003. 151–158
4 Theobald M, Schenkel R, Wiekum G. An efficient and versatile query engine for TopX search. In: Proceedings of the 31st International Conference on Very Large Data Bases, New York, 2005. 625–636
5 Beigbeder M, Gery M, Largeron C, et al. ENSM-SE and UJM at INEX 2010: Scoring with Proximity and Tags Weights. Berlin Heidelberg: Springer, 2011. 44–53
6 Metzler D, Croft W B. A Markov random field model for term dependencies. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, 2005. 472–479
7 Clarke C L A, Cormack G V, Tudhope E A. Relevance ranking for one to three term queries. Inform Process Manag, 2000, 36: 291–311
8 Peng F, Ahmed N, Li X, et al. Context sensitive stemming for web search. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, 2007. 639–646
9 Song R, Taylor M J, Wen J R, et al. Viewing Term Proximity from a Different Perspective. Berlin Heidelberg: Springer, 2008. 346–357
10 Svore K, Kanani P H, Khan N. How good is a span of terms? exploiting proximity to improve web retrieval. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval,

New York, 2010. 154–161
11 Rasolofo Y, Savoy J. Term Proximity Scoring for Keyword-Based Retrieval Systems. Berlin Heidelberg: Springer, 2003. 207–218
12 Gao N, Deng Z H, Jiang J J, et al. MAXLCA: a new query semantic model for XML keyword search. J Web Eng, 2012, 11: 131–145
13 Gao N, Deng Z H, Yu H, et al. ListOPT: learning to Optimize for XML Ranking. Berlin Heidelberg: Springer, 2011. 482–492
14 Guo L, Shao F, Botev C, et al. XRANK: ranked keyword search over XML documents. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, New York, 2003. 16–27
15 Xu Y, Papakonstantinou Y. Efficient keyword search for smallest LCAs in XML databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, New York, 2005. 527–538
16 Liu Z, Walker J, Chen Y. XSeek: a semantic XML search engine using keywords. In: Proceedings of the 33rd International Conference on Very Large Data Bases, 2007. 1330–1333
17 Bao Z, Ling T W, Chen B, et al. Effective XML keyword search with relevance oriented ranking. In: Proceedings of IEEE 25th International Conference on Data Engineering, Shanghai, 2009. 517–528
18 Geva S, Kamps J, Lethonen M, et al. Overview of the INEX 2009 Ad Hoc Track. Berlin Heidelberg: Springer, 2009. 16–51
19 Itakura K Y, Clarke C L. University of Waterloo at INEX2008: Adhoc, Book, and Link-the-Wiki Tracks. Berlin Heidelberg: Springer, 2009. 132–139
20 Liu J, Lin H, Han B. Study on reranking XML retrieval elements based on combining strategy and topics categorization. In: Proceedings of INEX, 2007. 170–176
21 Mills T C. Time Series Techniques for Economists. Cambridge University Press, 1990
22 Rijsbergen C J. Information Retireval. London: Butterworths, 1979
23 Wilcoxon F. Individual comparisons by ranking methods. Biometrics Bull, 1945, 1: 80–83
24 Mann H B, Whitney D R. On a test of whether one of two random variables is stochastically larger than the other. Ann Math Stat, 1947, 18: 50–60
25 Abdi H. The Bonferroni and sidak corrections for multiple comparisons. Encyclopedia meas stat, 2007. 103–107
26 Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J Roy Stat Soc B, 1995: 289–300
27 Shannon C E. Prediction and entropy of printed English. Bell Syst Tech J, 1951, 30: 50–64
28 Yu J X, Qin L, Chang L. Keyword search in relational databases: a survey. IEEE Data Eng Bull, 2010, 33: 67–78
29 Li J, Liu C, Zhou R, et al. Top-k keyword search over probabilistic XML data. In: Proceedings of IEEE 27th International Conference on Data Engineering, Hannover, 2011. 673–684
30 Wang G, Yuan Y, Sun Y, et al. PeerLearning: a content-based e-learning material sharing system based on P2P network. World Wide Web, 2010, 13: 275–305
31 Bao Z, Lu J, Ling T W, et al. Towards an Effective XML keyword search. IEEE Trans Knowl Data Eng, 2010, 22: 1077–1092
32 Qin L, Yu J X, Chang L. Computing structural statistics by keywords in databases. IEEE Trans Knowl Data Eng, 2012, 24: 1731–1746
33 Li G, Li C, Feng J, et al. SAIL: structure-aware indexing for effective and progressive top-k keyword search over XML documents. Inf Sci, 2009, 179: 3745–3762
34 Feng J, Li G, Wang J, et al. Finding and ranking compact connected trees for effective keyword proximity search in XML documents. Inf Syst, 2010, 35: 186–203
35 Liu Z, Chen Y. Differentiating search results on structured data. ACM Trans Database Syst, 2012, 37: 4