一种通过内容和结构查询文档数据库的方法*

王晓玲[1+], 文继荣[2], 栾金锋[1], 马维英[2], 董逸生[1]

[1](东南大学计算机科学与工程系,江苏 南京 210096)

[2](微软亚洲研究院,北京 100080)

A Method to Query Document Database by Content and Structure

WANG Xiao-Ling[1+], WEN Ji-Rong[2], LUAN Jin-Feng[1], MA Wei-Ying[2], DONG Yi-Sheng[1]

[1](Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

[2](Microsoft Research Asia, Beijing 100080, China)

+ Corresponding author: Phn: 86-25-6896725, E-mail: wxling@yahoo.com

http://www.seu.edu.cn

Received 2002-04-04; Accepted 2002-10-17

Wang XL, Wen JR, Luan JF, Ma WY, Dong YS. A method to query document database by content and structure. *Journal of Software*, 2003,14(5):976~983.

http://www.jos.org.cn/1000-9825/14/976.htm

Abstract: Structured documents are made up of a few logical components, such as title, sections, subsections and paragraphs. The components in each structured document can be represented by an ordered tree model, which can also be viewed as a hierarchical concept relationship. To meet the user's requirements for more precise and concentrated search results, the retrieval techniques should allow the user to retrieve document components with varying granularity. This paper presents a method to query document database by content and structure. The key idea is to construct a more comprehensive similarity function by taking advantage of the inherent hierarchical structure in documents. This work combines Information Retrieval techniques, semi-structured data query and proximate search for document documents. The proposed method is evaluated on the Encarta encyclopedia document set and the experimental results show that it can provide more accurate and focused answers than traditional document retrieval methods.

Key words: document database; information retrieval; passage retrieval; structured document

摘 要: 文档是有一定逻辑结构的,标题、章节、段落等这些概念是文档的内在逻辑.不同的用户对文档的检索,有不同的需求,检索系统如何提供有意义的信息,一直是研究的中心任务.结合文档的结构和内容,对结构化

---

* This work was performed while the first author was a visiting student at Microsoft Research Asia.

WANG Xiao-Ling was born in 1975. She is a Ph.D. candidate at the Department of Computer Science, Southeast University. Her current research interests include database theory and XML. **WEN Ji-Rong** is a researcher in Microsoft Research, China. His research interested areas are batabase theory and information retrieval. **LUAN Jin-Feng** was born in 1974. His research interested areas are artifical intelligence and communication. **MA Wei-Ying** is a researcher in Microsoft Research, China. His research interested areas are data mining and multimedia management theory. **DONG Yi-Sheng** was born in 1940. His current research interests include database theory and informaiton process.

文件的检索,提出了一种新的计算相似度的方法.这种方法可以提供多粒度的文档内容的检索,包括从单词、短语到段落或者章节.基于这种方法实现了一个问题回答系统,测试集是微软的百科全书 Encarta,通过与传统方法实验比较,证明通过这种方法检索的文章片断更合理、更有效.

关键词: 文档数据库;信息检索;段落检索;结构化文档

中图法分类号: TP311 文献标识码: A

Document database received more and more attention because of their multiple applications in the areas such as digital library, dictionaries, encyclopedias, etc. With the wide use of XML[1], which is a standard format for WWW data exchange and transformation, the whole web can also be viewed as a large document database. Traditional document retrieval techniques normally concentrate on the content part and various words matching approaches are used to obtain relevant documents according to user queries[2]. How to employ structure information to enhance document retrieval is a new challenge for researchers.

On the other side, traditional information retrieval techniques treat each document as an atomic unit and return the whole documents to the user. But, in many cases, only a part of the document is relevant to the user's information need. The user has to scan each (usually very long) document to look for relevant answers. Passage retrieval is one of the techniques aiming to retrieve and return more compact and shorter answer to the user. Most previous work suggests using roughly fixed length passages, which may decrease retrieval performance due to discarding semantic relationships among the components in documents. In recent years, many models have been developed to search documents by combining content and structure[2,3]. How to retrieve the volume of structured documents more efficiently and return a more compact and precise answer to users gain more and more attentions.

In this paper, we propose a novel method to retrieve components of structured documents more accurately, which can be viewed as a question-answer system. Compared with document retrieval, the main task of a question-answer system is to provide a short and direct answer to a user query[4]. Our work focuses on helping users to locate the most matching answer from the underlying structured document collection. The experimental results show that our method can produce more accurate results and shorter answers than traditional document retrieval and, at the same time, provide much more related context information about fuzzy questions so that users can understand the answer better.

This paper is organized as the following: Section 1 is about related work about structure document retrieval. Section 2 defines some data structure and techniques for structural document retrieval and ranking. Experiments and evaluation of the proposed method is given in Section 3. Then, in Section 4, we will draw some conclusions and discuss future work.

1 Related Work

In the information retrieval area, there are mainly two kinds of structured document retrieval models-non-overlapping list and proximal node. For models based on non-overlapping list, each list is a segment of flat texts and lists are not nested. So it is easy to employ AND, OR and NOT operations of the Boolean model for retrieval. The results are a set of non-overlapped text segments without contexts. Proximal node models use a hierarchical structure to index document text. A typical indexing structure is a strict hierarchy, which is made up of nodes such as chapters, sections, paragraphs and pages. Each node is associated with the corresponding text segment. Recent years, there are also some related works on sub-string retrieval[3], and they have gained promising results compared with document retrieval. But most of those works focus on providing a fixed-length sub-string, such as 50~200 words, in order to avoid the normalization problem of the length of sub-string. They consider each query term is a start of new passage, so the total document can be many passages rather than the natural paragraphs

denoted by the author. As a result, semantics inherent in the natural paragraphs may be lost.

Another work about structured document retrieval is related to XML document retrieval. XML document also is called semi-structured document, which is different from both unstructured flat document and structured data. Recently, there are many endeavors to develop more efficient and effective methods for XML document retrieval. Some related works include XML document indexing and XML query language such as Lore[5,6] and XML Query[7]. But most of these works require artificial query languages for facilitating effective retrieval and few deals with the issue about using natural language to search XML documents.

There are some proposed passage definitions based on document markup, such as sections, paragraphs, group of sentences and fix-length sequences of words which can be either disjoint or overlapping[2]. But, direct passage retrievals do have one serious drawback: if they are naively implemented, the cost of passage indexing and ranking is high since the number of candidate passages in a collection is much larger than the number of candidate documents[2]. We will exploit a pre-filter method in this paper to condense the candidates and thus make the query processing efficient enough to be accessed on-line.

2 Retrieve Relevant Components from Structured Documents

In this section, we describe a novel method to retrieve most suitable components from structured documents. A new similarity function is proposed to improve matching precision and provide more compacted answer to users by taking advantage of the inherent structure information in documents. Moreover, natural language or keywords are used to represent users' information needs and no artificial query language is required. There are three basic assumptions behind our method:

1. Users are looking for more accurate and focused answers to their questions.

We observe that most answers for users are about a part of an article rather than the entire document. Because of this, the retrieval system can only return a part of the document to answer users' questions and thus users can avoid scanning the entire document to look for the answer.

2. There are inherent structures embedded in documents, such as chapter, section, sub-section, and paragraphs. And each structure has its specific content.

There are some independent semantic in each paragraph in many documents. Typically, a document can be viewed as a concept tree and different concept level is made up of title, section, and paragraph, etc.

3. If a term with low IDF is frequently appeared in a block of documents, it will also appear in that document's term list.

This assumption will guarantee that, generally, filtering articles by question terms will not miss those promising answers.

Our proposal method is implemented under the consideration of these three assumptions. The experiments in Section 4 also testify these assumptions.

The architecture of our proposed method is illustrated in Fig.1. Basically it is a two-layer architecture. The first layer is similar with traditional document retrieval system. The second layer is a novel paragraph retrieval system. When a user query is submitted, the document retrieval system retrieves a set of documents with high relevance. This document set is then sent to the paragraph retrieval system. The paragraph retrieval system select most relevant document components such as paragraphs, sections or any other logical structure and return them to the user. Below we will introduce these two parts respectively.

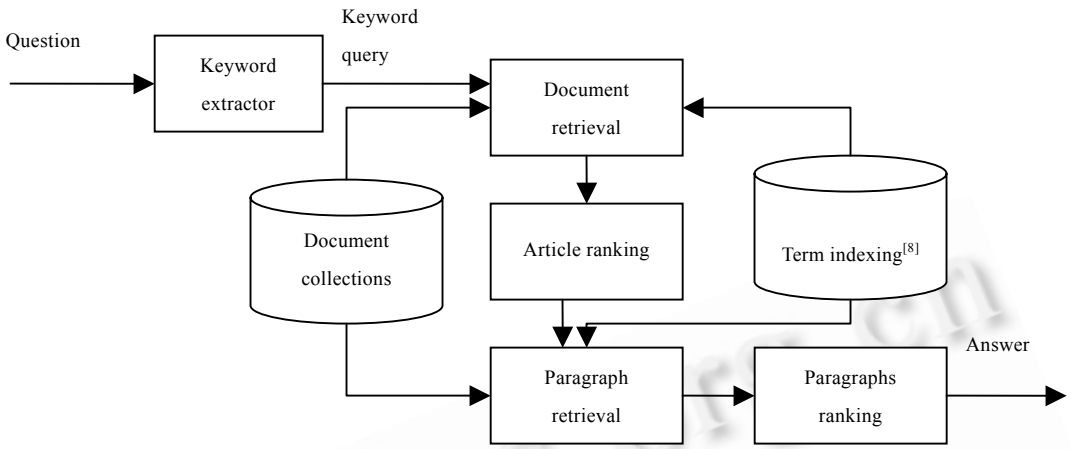

Fig.1 Document retrieval system architecture

2.1 Document retrieval and ranking

Traditionally, information retrieval mostly refers to document retrieval. Document retrieval mainly consists of building up efficient indexes, processing user queries with high performance and developing ranking algorithms, which improve the 'quality' of the answer set. Each document is treated as a sequence of words. And each query Q can be decomposed into a set of word terms: $Q=\{q_1,q_2,\ldots\}$. The assumption for tradition text retrieval is the similarity between document and query, and the weight associated with (word, document) quantifies the importance of the term for describing the document semantic contents[4]. For a query Q and a document D, their relevance measure is defined and computed by TF-IDF as the following:

$$Wt = tf * idf = tf * \log(N + 0.5) / n$$

$$Weight = \sum_{t=1}^{m} Wt;$$

where Wt is weight of query term t;

tf is the frequency of term t in document D;

idf is the inverted document frequency;

N is the document number of collection;

n is the number of document where term appears.

Inverted file is usually used as an indexing structure to facilitate document retrieval. The term file is described as $\langle term, document\_ID, frequency\rangle$ and the inverted file is $\langle term, idf\rangle$. Term weight Wt for term t in document D can be computed by getting frequency tf from term file getting idf from inverted file.

Based on this formula, all documents compute their relevance to the query terms and rank these values to get the top N candidate documents, which is an initial document collection for later paragraph retrieval.

2.2 Paragraph retrieval and ranking

In a typical information retrieval setting, a search for some query terms identifies documents containing all keyword "close" together, and a document is considered a "better" match if the keywords are near each other in the document text. The same idea is applied for paragraphs retrieval, that is, the nearer of query terms in some paragraphs, the higher weight of these paragraphs.

Taking into account the paragraph length, distance and the weight of query term in paragraphs, we re-definition term weights in a paragraph. Except the traditional keyword index for collections, there is another enhanced inverted file[8] for paragraphs retrieval. For each document, there is an inverted file to record the word position information and occurrence information in every document. The formal description is ⟨*word*, *frequency*, *positio*⟩*n*, where '*position*' is a structure (*section_ID*, *para_ID*), and *section_ID* can be nested, so it can be viewed as a hierarchy structure. The hierarchy in document allows us to measure the conceptual distance between paragraphs in a document. Below we will define such a king of distance. This definition is critical for finding the shortest sub-tree for a query.

Because the distance assignments must be made with a good understanding of the semantics, we define all children of the same parent is ordered from left to right and they are numbered is from 1 to n, so the left children is earlier than the right. The distance between two children for a same parent is the order of right child minus the order of the left child.

The distance is formally defined as following. '*x*' and '*y*' are elements in the same level, for example, *A* is in the level 0; *B*, *C*, and *E* are in the level 1; any words from the text of the node *D*'*s* content are in the level 3.

1. $D(x,y) > 0$;
2. $D(x,y) = D(y,x)$;
3. $D(x,x) = 0$;
4. $D(x,y) = 0$ if *x* and *y* are all elements of leaf node;
5. $D(x,x+1)$= the number of the children of node *x*.;
6. $D(x,y)$=abs(order(*x*)−order(*y*)) if x and *y* are elements of leaf nodes which are siblings but there are $n-1$ sibling between them, order(*x*) is the order of *x* in his sibling;
7. $D(x,y) = \sum_{i=x}^{y} D(i,i+1)$, if node *y* is later then node *x* in the text stream.

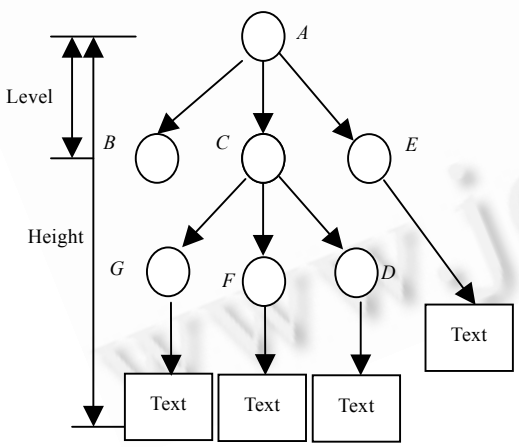

Fig.2 Document hierarchy structure

Thus, a distance function is defined and can be used to computer the distance between any two objects, including tree nodes and minimum elements decomposed from leaf node content. In Fig.2, $D(B,E)=5$; $D(G,F)=1$. If node *G* and node *F* have the highest weight for answering question, then this system will return the string sequence of node *G* and node *F*, rather than return node *C*. The weigh of query term in paragraphs is computed as the following delta weight:

$$delta\_Weight(t,i,d) = \sum_{i=1}^{n}\left(\sum_{t=1}^{m} Wi(t)\right) \Big/ n$$

$$(t) = tf * idf + tf / r = tf * \log(N + 0.5) / k + tf / r$$

where *m* is the number of query terms;

delta_Weight(t,i,d) is the weight of term *t* in the paragraph *i* of document *d*;

n is the number of the paragraph in the candidate result—subpart of the document *d*, i.e., the distance from the begin paragraph to end paragraph in each candidate result, and it is computed by the distance formula defined above;

N is the paragraph number in the document *d*;

k is the number of paragraph where query term t appears;

tf is the frequency of term t in the paragraph i;

r is the number of word in the paragraph i.

2.3 Query processing

Based on the definitions of weight and delta weight, the procedure of paragraph retrieval is described as the following:

1. Getting query terms' information from extended term index of each article.
2. Combining the paragraph lists generated by each query term and some paragraphs are selected as candidates.
3. For each document:
   a) ranking candidates by the number of query term appeared in each candidate.
   b) ranking candidates by delta weight.
4. Choosing the best candidate from each document.
5. Ranking candidates.
6. Getting the top 10 results.

In Step 6, a threshold is set to control the length of the result. Thus we can control the document components with varying granularity. The threshold is defined as (*weight*1−*weight*2)/*weight*1. The experimental results in Section 4 prove that the lower the threshold is, the more focused the answer is. Our experiments also suggest 0.1 is good value for tradeoff between match precision and coverage.

3 Evaluation

For evaluating our method, some experiments are conducted on the Encarta encyclopedia (http://encarta.msn.com). 22 queries collected from kids are used to retrieve more than 40,000 Encarta documents and the top 10 results are evaluated.

The first experiment is to test four ranking methods: article weight, paragraph weight, combined article and paragraph weight, and the method that directly comes from the Encarta website.

The five measure points are: 1—the precision of top 1 answer; 2—the precision of top 2 answers; 3—the precision of top 3 answers; 5—the precision of top 5 answers; 10—the precision of top 10 answers. By By By

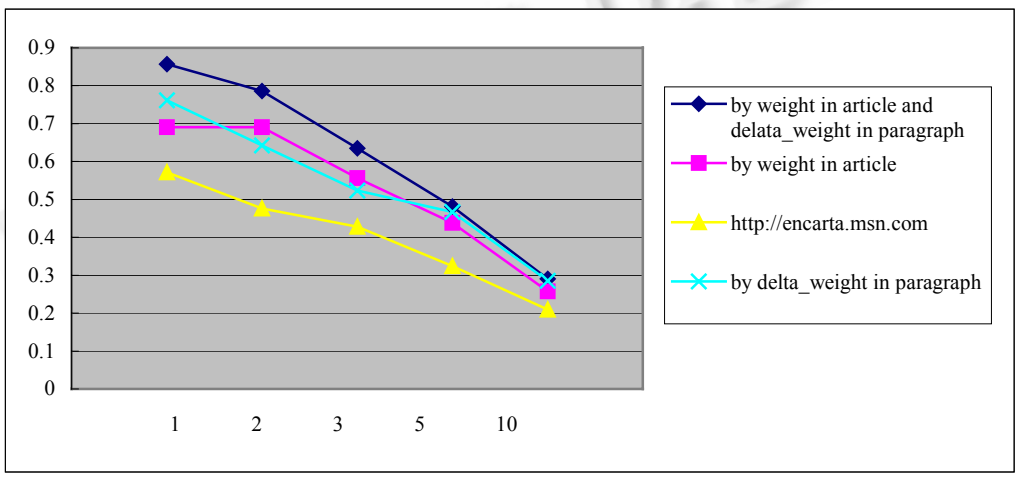

Fig.3 Performance of the four methods

Figure 3 illustrates the performance comparison of these four methods. The average number of relevant documents for each query is 62/22=2.8. So, the curve declines sharply when the *X* axis's value is larger than 2. If answers exist, in the top third results, paragraph retrieval can get one. Meanwhile, paragraph retrieval and traditional retrieval can get some text, which contains answer, but title retrieval doesn't work well for looking for answers. According to this figure, we can draw the following conclusions:

1. Combination of both article and paragraph weights can lead to higher precision for document retrieval.
2. Usually, using title alone cannot get satisfied results for question answer.
3. To retrieve relevant documents only by paragraphs can't always perform well.

The second experiment is set to test the precision of paragraphs retrieval and the experimental results are shown in Fig.4. The average number of relevant results for each query is 54/22=2.5. So, the curve declines sharply when the X axis's value is larger than 2. That is to say, most questions can get answer from the first three answers. We can easily find that ranking by combination of weight in article and *delta_weight* in paragraph can get higher precision than ranking only by *delta_weight*.

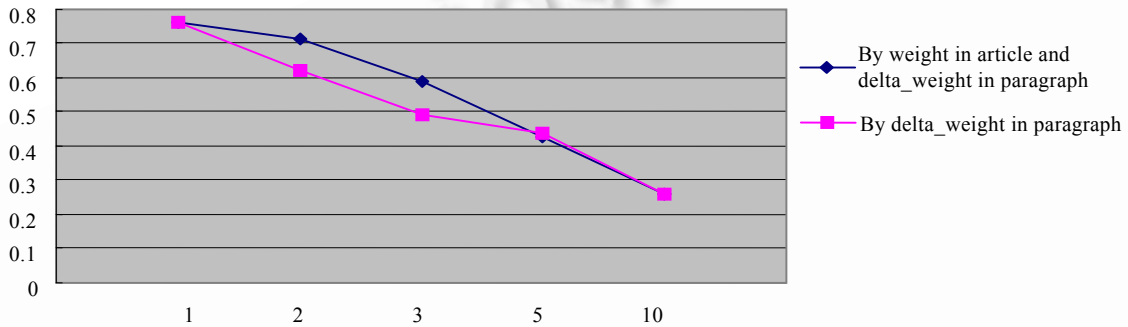

Fig.4 Comparison of the two paragraph ranking methods

4 Conclusion and Future Work

This paper put forward a new method for approximate search by taking advantage of the natural hierarchical structure in text documents. This method is implemented in a question answer system. Experiments show that the structure information is helpful for question answer system to suggest relevant answers better than traditional document retrieval. Paragraphs retrieval, which returns compact or relevant answers, is more reasonable than shortest sub-string retrieval and document retrieval.

The proposed paragraph retrieval method needs further studies. A better result representation with a hypertext structure would help much in practice. We also noticed that terms in specific document parts, e.g. title, are more important and could lead to good results for some queries. Thus, how to combine paragraph retrieval and title retrieval would be an interesting research topic. Another future works include improving efficiency, exploring tradeoff between the size of paragraphs and precision, and evaluating performance on some standard data sets, such as TREC.

Acknowledgement Thank Dr. Liu Wenyin and other researchers in MSR Asia for their valuable discussions and comments.

References:

[1] Extensible Markup Language (XML). http://www.w3c.org/XML/.

[2] Kaszkiel M, Zobel J, Sacks-Davis R. Efficient passage ranking for document databases. ACM Transactions on Information System, 1999,17(4):406~439.

[3] Clarke CLA, Cormack GV. Shortest-Substring retrieval and ranking. ACM Transactions on Information System, 2000,18(1):44~78.

[4] Cooper RJ, Rijger SM. A simple question answering system. In: Proceedings of the TREC-9. NIST Special Publication, 2000. http://www.doc.ic.ac.uk/~srueger/index.html.

[5] McHugh J, Widom J. Query optimization for XML. In: Proceedings of the 25th International Conference on Very large Data Bases. Edinburgh, Scotland, 1999. 315~326.

[6] Goldman R, McHugh J, Widom J. From semistructured data to XML: Migrating the lore data model and query language. In: Proceedings of the 2nd International Workshop on the Web and Databases (WebDB'99). Philadelphia, 1999. 25~30.

[7] XML query. http://www.w3c.org/XML/Query.

[8] Wang XL, Wen JR, Liu WY, Dong YS. Enhancive index for structured document retrieval. In: Proceedings of the12th International Workshop on Research Issues on Data Engineering: Engineering E-Commerce/E-Business Systems (RIDE-2EC 2002, Workshop of ICDE 02). San Jose, California: IEEE, 2002. 34~38.