Research on XML-based Information Retrieval Model

Yanping Wang, Kuanjiu Zhou
School of Management, Dalian University of Technology, Dalian 116024, China

ABSTRACT

Traditional information retrieval model is based on the statistics, and all the keywords to the model are not assigned with weight values. So the precision of retrieval results is not high enough to satisfy users. A new information retrieval model is issued to evaluate the weight and similarity by adopting influence factors of keywords in XML documents and context knowledge. It is applied to an information retrieval system (XIIR) for domain-based XML documents retrieval. The experimental results show that the model is more effective in raising precision and recall rate than traditional models. At last, the development of information retrieval model is prospected.

Keywords: XML, Information Retrieval, Vector Retrieval Model, Keywords Weights, Domain knowledge

1. INTRODUCTION

On the condition of the Internet, most of the documents exist in HTML form. Their tags are translated from HTML to XML gradually[1]. From the point of view of technology, the origin of a text logic section with a certain feature can be recognized with XML. Each characteristic vector and weight vector can be carried out with text-processing automatically, so the vector retrieval model of file-structure-based can be implemented smoothly. Due to adopting unified mode to express heterogeneous data, XML makes it possible to exchange data between different realms. There is a lot of study in information retrieval model about XML-oriented text. Theobald and Weikem[2] designed XXL query language and corresponding retrieval method to offer imprecise matching by adopting similar comparison into XMLSQL and to compute the correlative degree between text and query. Fuhr and Grobjohann[3] analyzed the application environment of XML and brought forward XIRQL query language in which data and text retrieval were associated, and then they added the information retrieval feature to XML documents in a fuzzy logic model to get the related probability. Hayashi[4] and others researchers designed an XML retrieval system and implemented a domain text retrieval system by restricting information structure index and query information structure. Hatano[5] and other researchers designed an information retrieval system in which child documents of a XML–based document were regarded as retrieval results. All the systems improve the precision and recall rate to some extent. Some practical applications prove these models are efficient.

All conventional information retrieval models are based on the statistics, and weight value is not attached to keywords flexibly in these models, so the precision is not high enough to satisfy user's demand. This paper presents a context knowledge-based retrieval model to satisfy the increasing demand from users. This model is applied to an information retrieval system prototype (XIIR) implemented by our group running on windows2000 platform. The system optimizes the user retrieval statement at first and then to parse the XML data source, at last it sorts the retrieval results.

2. THE KEYWORD WEIGHTING MODEL

There are Boolean Model, Vector Space Model, Fuzzy Logic Model and Probabilistic Model etc in information retrieval model researches. The Vector Space Model (VSM)[6] translates the matching of text information into the vector matching of vector space. It is regarded that a text is made up of separate vocabulary entry groups (T1, T2, … , Tn). T1, T2, … , Tn can be regarded as

coordinate axis of a coordinate frame with n dimensions, the related weights W1,W2, … ,Wn which reflect their importance degree are regarded as coordinate values. All the queries and documents can be mapped to the text vector space. Let's suppose user query as Q and retrieved text as D, and then the similarity could be measured by the cosine value of the angle between the two vectors. The formula of the similarity is shown as follows:

$$sim(Q,D)=\cos(Q,D)=\frac{\sum_{k=1}^{n}W_{qk}\times W_{dk}}{\sqrt{\sum_{k=1}^{n}W_{qk}^{2}}\times\sqrt{\sum_{k=1}^{n}W_{dk}^{2}}} \qquad (1)$$

Here, the weight in the formula is computed with conventional TF-IDF[6]. The difference between two documents is reflected by the cosine value of angle between two vectors. The problem is that useful words in one document are only a fraction of the total words and most of the words in the document are useless for assorting. Now, two methods are issued to improve the demerit.

(1) Weighting of keywords in data tree

The value of weight is confirmed according to the importance of the word. During a retrieval, every node in the XML data tree is endued with a weight coefficient *a* ($0\leqslant a\leqslant 1$) to ensure that the sum of weight coefficients of all the nodes is 1, and each node's weight coefficient is fixed according to the comparative importance of signification of all the child nodes for the node.

By this token, the value of primary TF can be modified with considering the influence of the positions where the keywords appear in the data tree. So the formula can be modified as follows.

$$\mathrm{TF}_{k,v}=\sum_{v_d\in f(v)}(\mathrm{TF}_{k,v_d}\times Weight(v_d)) \qquad (2)$$

TF_{k,v_d} -- The value of TF that the keyword k is in the node v

$\mathrm{TF}_{k,v}$ -- The value of TF with the influence of weight coefficient.

$$Weight(v)=\prod_{v_p\in path(root(T_c),v)}a_{v_p} \qquad (3)$$

root（T_c）--The root node of the child tree T_c, $path(root(T_c),v)$ --The path from the root node of T_c to node v

v_p -- All the nodes on this path

a_{v_p} --The weight coefficient of node v_p.

In one text, the subject is more accurate than chapter to reflect the topic. According to this, different nodes in XML tree can be endued with different weight coefficient to judge the importance of nodes. For instance, the weight coefficient of child node "subject" is higher than that of "chapter", and the weight coefficient of child node "number" is endued with 0 because it has no information, and node "subject" is endued with higher value in node "chapter". The weight from the formula is absolute. It must consider the influence of different positions of keywords to analyze the comparative weight, so the absolute weight should be used by the formula (2) to compute the comparative weight.

(2) Context knowledge weight tree

Because one word may have several meanings, the specific can be confirmed in certain context, then the context knowledge database is set up to store the related context knowledge and its weight coefficient in one domain. While searching, the related context is amplified from the original keywords, and new retrieval words are gained from the weight coefficient of each word. The result is more sincere to reflect user's requirement. XIIR is a system that orients one domain, so a knowledge database is built to store the useful domain vocabularies, and it is called context knowledge database. The weight b (b>0) of vocabulary can be initialized based on its serviceability, while searching, if the keywords to be searched exist in the knowledge database, the weight can

be adjusted based on their weight in the knowledge database. The related weight tree is called context knowledge weight tree.

The final formula of weight coefficient is shown as follows.

$$\text{weight} = \frac{\sum_{i=0}^{n}(weight_i \times words)}{\sum_{i=0}^{n}(weight_i)} \quad (4)$$

Here, we sum up the two weighting methods, and the final similarity formula is shown as the follows.

$$sim(Q,D) = \begin{cases} 1 & if \quad Q = D \\ \alpha_1 \times sim_{key}(Q,D) + \alpha_2 \times sim_{context}(Q,D) & if \quad Q \neq D \end{cases}$$

$$(\alpha_1 + \alpha_2 = 1, \text{and} \alpha_1 \geq 0, \alpha_2 \geq 0) \quad (5)$$

The similarity based on the importance of structure and context knowledge is adjusted. If their status is equal, then $\alpha_1 = 0.5, \alpha_2 = 0.5$, or if the context knowledge is the most important, then $\alpha_1 = 0, \alpha_2 = 1$

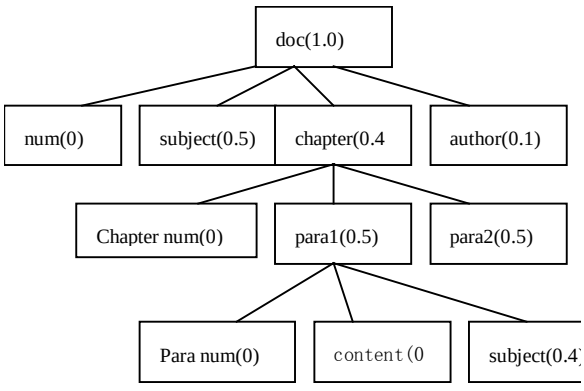

Fig. 1 The Tree of XML Document

For example, in Fig. 1, a document is regarded as a tree, and the weight coefficient of each keyword is also labeled. We can take this query tree as an example, the weight of the node “doc”- “chapter”- “para1”- “subject” is 0.4 ×0.5×0.4×1 = 0.08. Similarly, other weights of the keyword “subject” in this document can be computed out and we can suppose that they are: 0.5,0.7,0.09 … then the final TF can be gathered in formula (2):

TF（ “subject”）=0.08*TF(subject 1)+0.5* TF(subject 2)+0.7* TF(subject 3)+0.09* TF(subject 4)+… …

The result reflected by the keyword weighting can be computed by the above formula. According to TF-IDF, the result of the sum of $sim_{key}(\text{"subject"})$ can also be gotten.

For another example, in the context knowledge database, the weight of keyword “subject” is 2, the weight of keyword “subject’s context ” is 5, while users search the keyword “subject” in a domain knowledge database, weight is shown as follows.

Weight = （“subject” ×2＋“subject’s context” × 5）/(2+5)，it is the same while computing the similarity:

$$sim(\text{"subject"}) = 0.4 \times sim_{key}(\text{"subject"}) + 0.6 \times sim_{context}(\text{"subject"})$$

Because the effect of keyword is less important than that of context, the influence factor of keyword is confirmed as 0.4 and that of context is 0.6 in this formula.

3. INFORMAION RETRIEVAL SYSTEM

An information retrieval system XIIR that orients a domain is designed based on the improved model above. It is implemented with JBuilder9 and Microsoft SQL Server 2000[7] as storage platform. The information of XML is mined with DOM. At last, the storage of relational database is implemented with JDBC.

There are 3 steps to complete the retrieval:

(1) Filtrate the keywords input by user.
(2) Set up the context knowledge database and confirm the weight of keywords in the database.

(3) Parse the XML documents, construct corresponding document tree, and extract the label content of the nodes to match the user's query, then retrieval with the VSM.

The structure of the system is described in Fig. 2.

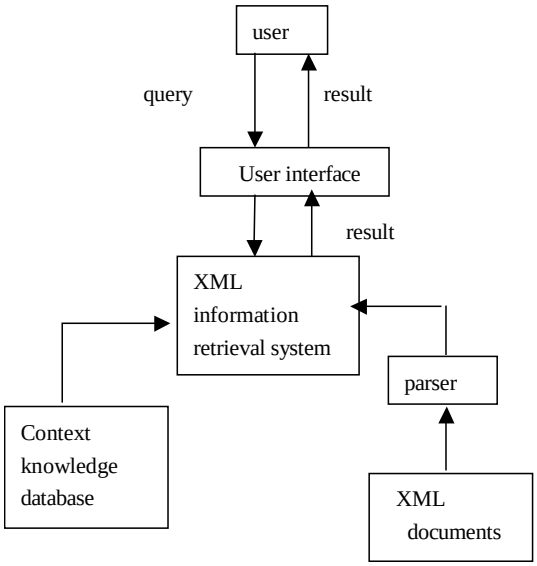

Fig.2 information retrieval model

Context knowledge database: there are domain vocabularies in the database, and they represent the core content and key point in this domain. It needs the knowledge of experts in this domain to confirm the weights of these words.

XML VSM retrieval system: a system is designed with the weighting model.

User interface: it is used for the interaction between user and the system and brings forward the query and decides whether to search again or not by user's satisfaction degree.

Parser: its function is to parse the XML documents and to construct corresponding tree by SAX or DOM, then extract the label content. When the query is represented, the system will match the content with user's query.

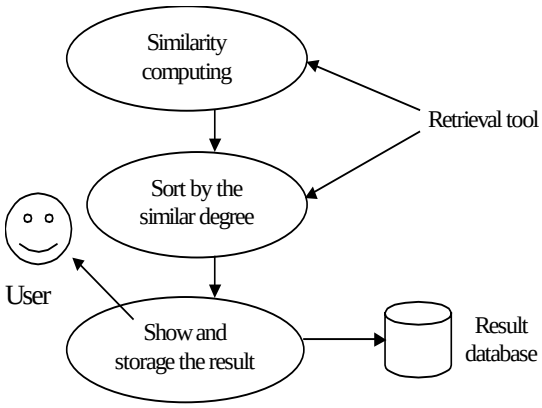

Fig. 3 Retrieval module

4. EXPERIMENTS AND RESULTS

4.1. Experiment Data

In order to verify the XML-based query, we make an experiment. In this experiment, we select the blending XML data that is made up of the drama of Shakespeare and religion books. The character of the data is that they both have complex structure and small amount. It is shown as follows.

(1) The religion books[8] include 4 documents, each document's size is about 1M-3M, and the amount of all elements in the documents is 48,259;
(2) There are 27 documents which size is about 300k in the drama[9] of Shakespeare, and the average amount of each elements in the documents is about 10,000;

4.2. The Query Experiment

There are 300 queries in this experiment, now we analyze the examples as follows.

The information requirements are shown as follows.

(1) Search the documents about "Bible";
(2) Search the role about "prince";

Construct the query as follows.

（1）Q1: title [Bible]

(2) Q2: person [prince]

The words amplified from query 1 are Biblist, Jesus, Christianism and so on and the words amplified from query 2 are princess, love and so on. Search them with the XIIR separately, and analyze the information with the top 10 similarities.

Similarly, there are experiments on other queries, and there is much data to compare the influence from weighting retrieval model.

Experiment 1: Test the average precision and recall rate with two kinds of retrieval methods, one is normal retrieval and one is XIIR, the data is in table 1.

Table1 the average precision and recall rate with two kinds of retrieval methods

Method	Average precision (%)	Average recall rate (%)
Normal retrieval	19.21	63.32
XIIR	24.24	72.80

Some conclusions can be drawn from the above statistical table: the precision and recall rate of the normal retrieval system are lower than that of the XIIR system. The average precision of the XIIR system is higher besides the average recall rate. The table shows that the new information retrieval model is feasible and more efficient and effective than traditional models.

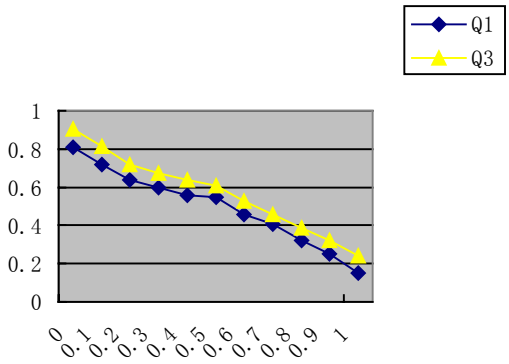

Fig.4 11 points of the precision and recall rate

Experiment 2: Test the precision and recall rate of 11 queries, the result is shown in graph 4, the x axis represents recall rate, and the y axis represents precision, Q1 represents normal retrieval and Q2 represents XIIR.

We can see from the result that when the retrieval model is added to query, the precision rises with the recall rate lifted. The reason is that the retrieval procedure is a sorting one, and the user can get the related result more quickly.

5. CONCLUSIONS

Due to fast increasing of information on the Internet, XML-based information retrieval model becomes hotspot[10-11] of research. In conventional information retrieval model, retrieval unit is full text, but for XML retrieval, each elements and attributes act as retrieval unit, the document is broken up into different hierarchy objects to describe the text structure more efficiently and clearly, then not only content but also structure should be considered to satisfy the user's demand. It increases the complexity of retrieval[12]. Therefore, a retrieval model based on weighting by context knowledge and keyword is issued, and an XML-based information retrieval system (XIIR) is designed to verify the rationality of this model.

REFERENCES

[1] Bray T , Paoli J , Sperberg McQueen C M , et al . Extensible Markup Language (XML) 1. 0 (Third Edition)[DB/ OL] . http :/ / www . w3. org/ TR/ 2004/ R EC xml ,2004202.

[2] Theobald A ,Weikem G. Adding Relevance to XML[A] .Proceedings of 3 rd International Workshop on Web and Database[C] . London : Springer2Verlag , 2000. 1052124.

[3] Fuhr N , Grobjohann K. XIRQL :A Query Language for Information Retrieval in XML Documents[A] . Proceedings of the 24 the Annual International Conference on Research and development in Information Retrieval [C] .New York : ACM Press , 2001. 1722180.

[4] Hayashi Y, Tomita J , Kikui G. Searching Text2rich XML Documents with Relevance Ranking [DB/ OL] .http :/ / www . haifa. il . ibm. com/ sigi r002xml/final2papers/Hayashi/hayashi.html ,2000207.

[5] Hatano K, Kinutani H , Yoshikawa M , et al . Information Retrieval System for XML Documents[A] . Proceeding of the 13th International Conference on Database and Expert Systems Applications [C] . London : Springer2Verlag , 2000. 7582767.

[6] Salton G, Wong A , Yang C S. A Vector Space Model for Automatic Indexing [A] . Communications of the A CM[C] . New York : ACM Press , 1975. 6132620.

[7] http://metalab.unc.edu/bosak/xml/eg/rel200.zip

[8] http://metalab.unc.edu/bosak/xml/eg/shaks200.zip

[9] David Carmel , YoÄlle S. Maarek , Matan Mandelbrod , Yosi Mass , Aya Soffer. Searching XML documents via XML fragments[A] . In Proceeding of ACM SIGIR2003 , Toronto , 2003.

[10] A. Schmidt , M. Kersten , and M. Windhouwer. Querying XML documents made easy : Nearest concept queries[A] .In ICDE , 2001.

[11] N. Fuhr , K. Grobjohann. XIRQL : A Language for Information Retrieval in XML Documents[A] . SIGIR Conf . ,2001.

[12] Lan Yan , Liu Tao , Luo Wei1A Knowledge Based Information Retrieval Systems KIRS Proceedings of International Conference on Information and Systems. AMSE ,1991