STRUCTURE-BASED QUERY EXPANSION FOR XML SEARCH ENGINE

Wei-ning Qian, Hai-lei Qian, Li Wei, Yan Wang and Ao-ying Zhou

Computer Science Department
Fudan University
Shanghai 200433
E-mail: wnqian@fudan.edu.cn

Abstract: Based on the query expansion techniques in information retrieval systems, structure-based query expansion for XML search engines, which is designed to ease the query for XML data while keeping the power and flexibility of XML query, is introduced in this paper. To enable the structure expansion, a structure thesaurus should be built first, which involves the construction of a weighted graph from XML documents and the linkage-based clustering method to cluster the nodes into several groups. After a query comes, the structure thesaurus is examined, so that for each tag in the original query, the tags in the same group are retrieved. Unrelated tags are filtered and some heuristic rules are applied to replacing the tags in the original query with the related tags and to expanding the structure. It is shown that using structure-based query expansion, the system can return result with high precision and recall.

1. INTRODUCTION

XML (Extensible Markup Language) is a specification of W3C (Bray, 1998). It is developed to complement HTML for data exchange on the Web. In recent years, XML has been more and more used in large information systems, such as digital libraries or information centers. In most of these systems, search engine is a major module.

XML search engine has gained its popularity over HTML search engine primarily due to two notable advantages it bears. 1) It provides the ability to query not only the content, but also the structure. 2) It usually has more complex and powerful query languages, such as XML-QL (Deutsch) and XQL (Robie, 1999). These languages allow users to query elements satisfying certain conditions. However, these two advantages of XML search engine also bring the following shortcomings: 1) It is difficult for users to pose accurate structure queries without knowing the schema of the XML data, which is difficult to obtain from a large, distributed XML repository. 2) Mastering the complex query languages remains a tough task for common users. In this paper, we apply query expansion techniques to structures to mask the complex query languages from the users.

Query expansion is widely used in information retrieval systems (Xu, 1996; Mandala, 1999) to increase the precision and recall. In recent years, it has been implemented in several famous search engines such as AltaVista (http://www.altavista.com) and Lycos

(http://www.lycos.com). However, traditional query expansion methods are designed for keyword-based queries and cannot fulfill the task mentioned above. Structure-based query expansion for XML search engine is designed to ease the query for XML data while keeping the power and flexibility of XML query. It helps to solve the following problems. Firstly, users may not write complex queries (regular path expression, etc.) and may only pose the query from one angle. However, the documents on the Web correspond to so many various DTDs that users can not browse, hence a simple query does not make sure the search engine find enough documents that are needed by users. Secondly, there are some tags in XML DTDs that have different names but are close to each other in a semantic sense and have similar context as well. Traditionally, only the tags that match the user's query will be considered and their sub-tags will be searched, while the similar tags that do not match the original query but may contain the information in need are neglected. By clustering the similar tags into groups, the approach we adopt to structure-based expansion will solve these two problems effectively and provide users with the information they require as complete as possible.

To enable the structure expansion, a structure thesaurus should be built first. Based on the analysis of the XML corpus, a weighted graph, in which nodes are tags and edges are tag/sub-tag relations, is constructed. Then, linkage-based clustering method such as ROCK (Guha, 1999) or CHAMELEON (Karypis, 1999) is employed to cluster the nodes into several groups. These groups constitute the structure thesaurus, since they are formed on the basis of the structure information. It is assumed that users only pose queries containing simple path expressions and simple value constrains. After a query comes, the structure thesaurus is examined, so that for each tag in the original query, the tags in the same group are retrieved. For each tag that is the target or that has some value constrains, the similarity between the sub-structure of the retrieved tags and that of the tag in the original query is computed. The unrelated tags are filtered to guarantee that the target can be reached while the constraints are still satisfied. Some heuristic rules are applied to replacing the tags in the original query with the related tags and to expanding the structure. It is shown that using structure-based query expansion, the system can return result with high precision and recall. In other words, it helps to fulfill the query task well, while keeping the advantages of XML.

The rest of the paper is organized as follows. The next section discusses the related work. Section 3 describes our approach to structure-based query expansion in detail. The figure of the system architecture along with a brief introduction of the architecture is included in section 4. Finally in section 5, concluding remarks and a discussion about the future work are offered.

2. RELATED WORK

We base our work on automatic query expansion, a technique widely used in information retrieval systems that is designed for dealing with the fundamental issue of word mismatch in information retrieval. Among a number of approaches to expansion, techniques that analyze the corpus to discover word relationships (global techniques) and those that analyze documents retrieved by the initial query (local feedback) has drawn much attention (Xu, 1996). As far as the tool in automatic query expansion is concerned, many kinds of thesauri have long been used such as thesaurus of different types and their combination in (Mandala, 1999).

In our approach, linkage-based clustering algorithm (Guha, 1999; Karypis, 1999) is employed to construct the structure thesaurus. Such algorithm is developed for data with boolean and categorical attributes to overcome the problems with the traditional clustering algorithms that use distances between points for clustering when dealing with data of this type. Linkage-based clustering algorithm involves a novel concept of links to measure the similarity between a pair of data points. Generally, the number of links between a pair of points is the number of common neighbors for the points and a pair of points can be defined as neighbors if their similarity exceeds a certain threshold. A kind of linkage-based clustering algorithm is ROCK (RObust Clustering using linKs) (Guha, 1999). It belongs to the class agglomerate hierarchical clustering algorithms, which begins with a set of points and each point is a separate cluster. The clusters with the maximum number of links are merged into one cluster and the process continues until the desired number of clusters is reached. Employing links as the main evaluation factor when merging clusters makes ROCK a robust algorithm that not only generates better quality clusters than traditional algorithms, but also exhibits good scalability properties.

XML search engine has some complex and powerful query languages, such as XML-QL (Deutsch, 2000). XML-QL combines XML syntax with semi-structured query language techniques. It uses path expressions and patterns to extract data from the input XML data; it has variables to which this data is bound; and it has templates which show how the output XML data is to be constructed. Both patterns and templates use the XML syntax. When restricted to relational-like data, XML-QL is as expressive as relational calculus or relational algebra. XML-QL data model is different from XML data model. The former assumes the semi-structured data model, in which data is represented as an edge-labeled graph, while the latter is better described as a node-labeled graph. Another important XML query language is XQL (Robie, 1999), which uses XML as a data model, and is very similar to XSL Patterns. XQL expressions are easily parsed, easy to type, and can be used in a variety of software environments - as part of a URL, in XML or HTML attributes, in programming language strings, etc. XQL has already been implemented in web browsers, document repositories, XML middleware, PERL libraries, and command-line utilities.

3. STRUCTURE-BASED QUERY EXPANSION

In this section, we describe how to apply query expansion techniques to structures in detail. We give the overview of our approach in the first subsection. Our goal can be achieved in three steps: mapping all of the XML documents to a weighted graph, constructing a structure thesaurus and query expansion based on it, which are presented respectively in three subsections.

3.1. Overview of Our Approach

Structure-based query expansion is proposed in order to increase the precision and recall of the query for XML data. We base our work on the following assumptions. Firstly, users only pose queries containing simple path expressions and simple value constrains. Secondly, users cannot browse XML DTDs and have no knowledge of the structure information about XML documents. Thirdly, traditional XML search engines only consider the tags that match the user's query, while neglecting all the other tags even though some of them are semantically similar to the tag in the original query. Structure-

based query expansion is a process of replacing the tags in the user's query with the tags that are highly related to the information required, which means unrelated tags should be filtered and all the tags left are those by which the information can be obtained directly and completely.

The completion of the structure expansion mentioned above depends on the XML documents' structure information rather than their content. Useful structure information can be extracted from all the XML documents on the Web by constructing a structure thesaurus, which is composed of groups of tags and tags in the same group are close to each other in a semantic sense and have similar context. A weighted graph reflecting the structure information of all the XML documents is generated first to facilitate the grouping of tags, which can be achieved by applying linkage-based clustering algorithm to the graph. Thus, the implementation of our approach can be divided into three phases: mapping XML documents to a weighted graph, the construction of the structure thesaurus and query expansion based on the structure thesaurus.

In the first phase, the XML corpus is analyzed and a weighted graph, in which nodes are tags and edges are tag/sub-tag relations, is constructed. The work in this phase will be introduced in greater detail in the next subsection. Then, in the second phase, linkage-based clustering algorithm is applied to the graph to generate the structure thesaurus. More details about this phase are presented in subsection 3.3. Based on the work done in the previous two phases, the third phase can be completed more efficiently. By examining the structure thesaurus, all the tags in the same group as the tag in the original query are retrieved, so that all the similar tags in the XML documents would be considered. This process is explained in subsection 3.4.

3.2. Mapping XML Documents to a Weighted Graph

Obtaining structure information is the prerequisite for constructing the structure thesaurus. To ease the work in the second phase, structure information should be organized as a weighted graph, in which nodes are tags and edges are tag/sub-tag relations. Therefore, all the XML documents should be mapped to a weighted graph first.

To begin with, the XML corpus is analyzed, that is, we scan each XML document. All the tags and relations are mapped into nodes and edges on the fly until all the XML documents have been scanned and all the nodes and edges constitute the graph we need. In addition to simply mapping each tag to a node in the graph, identical tags are merged, which means tags with the same name in different XML documents would appear only once in the graph. Furthermore, each edge in the graph is associated with a weight to represent the total times this relation appears in all the XML documents. The mapping algorithm is presented in Figure 1. It involves a stack whose top element is the start point of a new edge or an existing edge whose weight should be increased. Create_node(t) is a function that maps tag t into a node in the graph. Function Detect(j) returns a non-zero value to indicate the existence of the node representing tag j in the graph, and it returns zero if the node has not been created. The rationale of function add_edge(i,j) is as follows: if there is no edge from node i to node j, it adds an edge between them and set the weight of the edge to 1, and if there has already been an edge between them, it just increases the weight of the edge by 1.

```
procedure mapping()
begin
  while(there are XML documents that have not been scanned) do
  {  t=the first start tag in the XML document being scanned ;
     create_node(t) ;
     i=0 ;
     stack[i++]=t ;
     for each tag from the second tag the in the document do
     {  if(the tag is a start tag) then do
        {  j=starting tag ;
           if(!Detect(j)) then do
              create_node(j) ;
           add_edge(stack[i], j) ;
           stack[i++]=j ;
        }
        if(the tag is an end tag) then do
           i-- ;
     }
  }
end
```

Figure 1. Mapping algorithm

3.3. The Construction of the Structure Thesaurus

The construction of the structure thesaurus is a process of clustering the tags that are semantically close and have a similar context into a group. The structure thesaurus is the set of all these groups. A linkage-based approach should be employed, which captures the global knowledge of neighboring data points into the relationship between individual pairs of points. This approach involves an important concept: links between data points, the number of which is the number of common neighbors for the points. The notion of neighbor depends on where the approach is applied. Generally, a pair of points is defined as neighbors if their similarity exceeds a certain threshold.

We adopt ROCK-clustering algorithm (Guha, 1999). Here, the input is the set of all nodes in the weighted graph generated in the first phase. A pair of nodes is called neighbors if and only if there's an edge between them. The context of a tag mentioned above actually includes all of its neighbors and all the edges connecting to it along with the weight information. We still define link (p_i , p_j) in the algorithm to be the number of common neighbors between node p_i and node p_j , but the weight of each edge should be taken into consideration as far as their common neighbors are concerned, which means a node q can be called a common neighbor of p_i and p_j if and only if the difference between the weight value of the edge p_i-q and that of p_j-q does not exceed a predefined threshold. Thus, the number of the links between a pair of nodes actually reflects the degree of the similarity of their context, and nodes with more links, that is, have more similar context, are more likely to be merged into a single cluster. The details of the clustering algorithm are presented in (Guha, 1999).

3.4. Query Expansion Based on the Structure Thesaurus

On the basis of the preparations made in the previous two phases, we embark on expanding the query according to the structure thesaurus in this phase. Whenever a query comes, the structure thesaurus is examined, so that for each tag in the original query, the tags in the same group are retrieved. For each tag that is the target or that has some value constrains, the similarity between the sub-structure of the retrieved tags and that of the tag in the original query is computed. Tags with similarity lower than the threshold are considered as unrelated tags and should be filtered to avoid unnecessary efforts and guarantee that the target can be reached while the constraints are still satisfied. Some heuristic rules are applied to replacing the tags in the original query with the related tags and to expanding the structure. Because of the lack of space, we omit the details here.

4. SYSTEM ARCHITECTURE

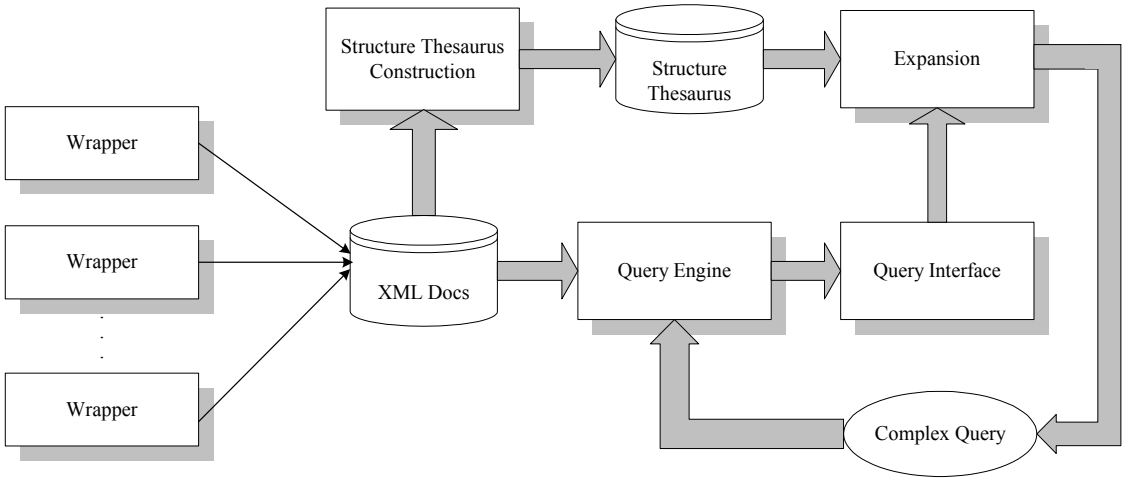

Figure 2: System Architecture

Structure-based query expansion is designed for XML search engines and should be integrated into the system to be employed by XML query engine. The system architecture presented in Figure 2 shows how structure-based query expansion is implemented in the system and how it cooperates with XML query engine to ease the query for XML data. All the XML documents are obtained from wrappers and constitute the base for constructing the structure thesaurus. The first two steps mentioned above are fulfilled by Structure Thesaurus Construction module, through which structure information is extracted from all the XML documents and the structure thesaurus is formed. Query Interface accepts the simple query submitted by the user and hands it over to Expansion module, which expands the original query according to the information in the structure thesaurus and generates a complex query. It is Query Engine that fulfils the searching tasks according to the complex query and returns the information required by the user through the interface.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we propose an approach to structure-based query expansion and displays how it is implemented and utilized in the XML engine system to ease the query for XML data while keeping the power and flexibility of XML query. The approach we adopt is oriented to queries containing simple path expressions and simple value constrains. It considers not only the elements in the original query posed by the user but also all the elements that are semantically close to them and have similar context when searching XML documents, hence it increases the precision and recall of the query. Besides its effectiveness in theory, the approach is feasible in practice. It successfully applies the clustering technique to the structure information of XML documents, which is extracted from XML documents by scanning all the XML documents on the Web and mapping them to a graph. Linkage-based clustering methods are available to fulfil the clustering task, which facilitate our work greatly. Once the structure thesaurus is constructed, it can be used by Query Expansion module for all the queries posed by users, which enables a through search for the information in need by XML query engine.

Our work is currently being extended in several directions, as illustrated below.

- Test the approach with different real-life data,
- Improve the current query expansion strategy to test whether the precision and recall can be increased,
- Use other linkage-based clustering algorithms to cluster the elements in the graph and test the result, and
- Construct different structure thesaurus to test the algorithm.

REFERENCES

Bray, T., J. Paoli, C.M. Sperberg-McQueen, and E. Maler. (1998). *Extensible Markup Language (XML) 1.0.* http://www.w3.org/TR/1998/REC-xml-19980210.

Deutsch, M., F. Fernandez, D. Florescu, A.Y. Levy, and D. Suciu. *XML-QL: A Query Language for XML*. http://www.w3.org/TR/NOTE-xml-ql/.

Guha, S., R. Rastogi, and K. Shim. (1999) "ROCK: A robust clustering algorithm for categorical attributes," *ICDE,* pp. 512-521.

Karypis, G., E.H. Han, and V. Kumar. (1999) "Chameleon: Hierarchical clustering using dynamic modeling," *IEEE Computer,* 32 (8): 68-75.

Mandala, R., T. Tokunaga, and H. Tanaka. (1999) "Combining multiple evidence from different types of thesaurus for query expansion," *SIGIR,* pp. 191-197.

Robie, J. (1999, March). http://www.ibiblio.org/xql/.

Xu, J. and W.B. Croft. (1996). "Query expansion using local and global document analysis," *SIGIR,* pp. 4-11.