社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－DBS－146 (20)
2008／9／22

異種型データにおけるキーワードによる問い合わせ処理について

キーワードによる問い合わせ処理への代数的なアプローチ

プラダンスジット†

† 倉敷芸術科学大学・産業科学技術学部 〒712–8505 岡山県倉敷市連島町西之浦 2640 番地
E-mail: †sujeet@cs.kusa.ac.jp

あらまし 異種型データにおいては，データの統一的な管理・シームレスサーチ手法が最も重要な課題と考えられる。これらの課題における多くの問題点の中の一つは，キーワードによる検索を行う際，“what is a suitable *retrieval unit*?”（何が適切な検索結果の単位であると）いう問題点が挙げられている。本論文では，この問題点を解決する場合に必要とされる要件や今後の研究方針について具体的に述べる。

キーワード 異種型データ, 情報単位, 検索結果単位, 問合せ代数

A General Query Model for Keyword Queries: Hopes and Challenges

Algebraic Approach to Keyword Queries

Sujeet PRADHAN†

† Faculty of Science and Industrial Technology, Kurashiki University of Science and the Arts
2640 Nishinoura, Tsurajima-cho, Kurashiki 712–8505 Japan
E-mail: †sujeet@cs.kusa.ac.jp

Abstract Keywords queries on various data models have been studied in the past. Regardless of the nature of target data, a keyword query poses a common problem: the difficulty in determining a suitable retrieval unit. In this paper, we discuss this problem in a new scenario that assumes several different types of data being stored and queried in a unified framework. A hypothetical general query model that borrows the ideas of our previous model designed for XML and linear video data is presented. We then state the requirements and challenges to realize such a model.

Key words heterogneous data, query algebra, information unit, retrieval unit

1. Introduction

Keyword queries in the past were limited to plain document search. However, recently they have been studied in the context of several other data models. The type of underlying data supported by these models range from simple linear video data [16] to more complex tree-structured XML data [14] [5] [11] [17] [21] or graph-structured data comprising web pages [10] [20]. Despite the basic structural differences, a keyword query over these various types of data poses a common problem: the difficulty in determining a suitable *retrieval unit* to answer the query. This problem is largely due to the fact that an *information unit* defined in the data set often fails to possess enough information in itself to be qualified as an answer to the given query. For example, as shown in Fig. 1, a keyword query `{sujeet, heterogeneous}` would fail to return any query result, as the query keywords are split across multiple nodes of a tree representing the XML data. Much work has been done to compute a suitable *retrieval unit* (the leftmost subtree rooted at `author` in Fig. 1) as accurately as possible [5] [11] [17] [21].

Moreover, even if an *information unit* does possess enough information in itself, it may not possess enough context in itself to be called a *good* (suitable) answer. For example, in [16] this problem was described in the context of linear video data. As shown in Fig. 2, a keyword query `{dog, man}` is more suitably answered by a set of consecutive shots from `s1` to `s4` rather than the single shot `s2` even though it con-

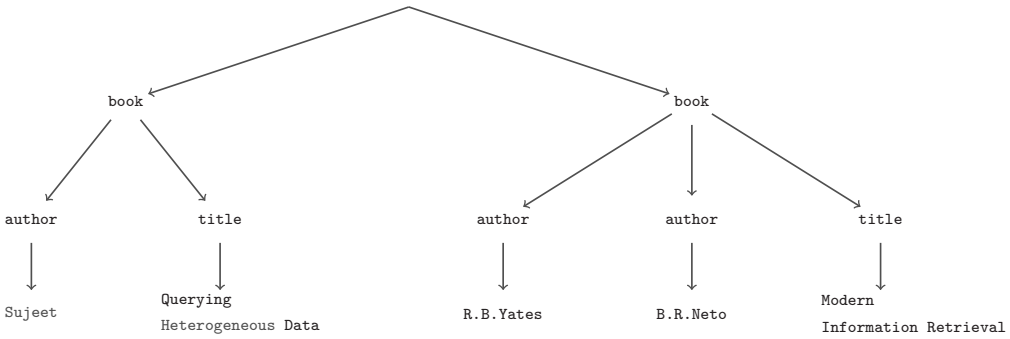

図 1 XML tree: each node representing an *information unit* (XML element)

tains both the query keywords ('dog' and 'man') because the shot, as it is, does not contain enough context in itself. [14] provides similar kind of arguments in the context of document-centric XML data.

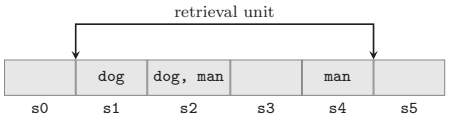

図 2 Linear video data: each shot representing an *information unit*

As seen in the examples above, a good *retrieval unit* to a keyword query often consists of several *information unit*s and this answer must be computed with the help of *additional information*. The nature of this *additional information* may vary depending upon the type of the underlying data that we are dealing with. For example, in the context of video data, temporal relationships among the *information unit*s can be used to generate a suitable *retrieval unit*. Similarly, structural relationships can become very handy for determining suitable *retrieval unit*s in the context of document-centric XML data, whereas in the context of data-centric XML data, both structural and semantic relationships can have enormous impact for the same task. Obviously, links provide excellent assistance in the context of Web data.

Data heterogeneity has emerged as one of the key issues in recent database applications. These applications demand several different kinds of data be stored and queried in a single unified framework. One such killer application is Personal Information Management Systems (PIMS) and some preliminary work have already been done in this area [6] [15].

Naturally, a keyword query over heterogeneous data poses the same problem regarding a *retrieval unit*. In a unified framework, however, the individual solutions provided for various types of data having structural and/or semantical differences cannot be applied directly to solve this problem. In this paper, we provide some insights for deriving a general query model for keyword queries over heterogeneous data. This model would be based on the results of our previous models that we developed for linear video data [16] and XML data [14]. Our goal is to investigate, (a) what the requirements would be and (b) how big the challenges would be, for achieving similar kinds of result in this new scenario. As this work is still at a preliminary stage, all of our discussions will be purely informal.

2. Related Work

Study of keyword queries over non-traditional data (other than plain text documents) is relatively new and we review below several contributions related to our work. We divide them into three categories: primary target data type, basic approach, and the semantics of a *retrieval unit*. For each of these categories, we discuss how our work would distinguish itself from earlier approaches.

Primary target data type: Keyword queries have been studied over several different data types; data-centric XML being the one most intensively focused upon [4] [5] [11] [17] [21]. Data-centric XML documents are highly schematic and their element tag names are generally "semantically meaningful". A great deal of effort has been put on to exploit both the schema and the tag names so that meaningful *retrieval unit*s are identified as precisely and concisely as possible.

On the other hand, studies described in [14] [2] [7] [8] [18] are focused on keyword search over document-centric XML documents. Most except [14] have been influenced by conventional IR-style query mechanism.

Another popular target has been relational data [9] [3] [1] [13] [12] mainly because more and more information stored in a database server are becoming available in the Web. The

main issue here is the efficient detection of a suitable *retrieval unit* consisting of several *information unit*s scattered across a set of relations.

One that is clearly out of line is our own work on linear video data [16]. This is also one of the earliest studies to identify the problem posed by a keyword query, which we stated in Section 1. To our best knowledge, none of the studies so far have made any attempt to deal with this issue in the context of heterogeneous data.

Basic approach to query processing: Two most notable approaches to processing keyword queries are (a) set-based approach [14] [16] [2] [17] and (b) navigational approach [10] [20] [9] [3] [1] [11]. In the set-based approach, a keyword query is treated as a collection of algebraic operations on a set of input data. Several operations need to be defined depending upon the type of input data and the semantics of the query output. Generally, set-based approach is considered more reliable and stable than other ad hoc approaches. Navigational approach, on the other hand, is more frequently taken especially when the input data is of nature having tree or graph-structure. Generally, a *retrieval unit* in a graph-structured data (tree is a special case of graph) is a subgraph and computing a subgraph with this approach is more intuitive than with the set-based one. However, navigational-approach lacks flexibility and cannot be easily extended. Our approach would be set-based approach, at least in the logical level, since applications dealing with heterogeneous data would demand not only for robustness but also for extensibility.

Semantics of a *retrieval unit*: There has been some discrepancies upon what a good *retrieval unit* is. Naturally, the semantics of a suitable *retrieval unit* depends upon the nature of data to be retrieved. For data-centric XML tree data, a *minimal subtree* containing all the keywords at least once seems good enough to be called a suitable *retrieval unit*. On the other hand, in [14] it was argued that this concept may not always be good enough when the target XML data are less schematic. It then provides a new semantics that goes beyond the concept the *minimal subtree*.

[10] is one of the earliest studies that states the problem of *retrieval unit* in the context of hyperlinked web pages. There is a recent study in the same line [20]. Both studies have graph-structured data as base model, and 'Steiner tree' is the main concept for determining a good *retrieval unit*. The concept of 'Steiner tree' has also been adopted for keyword query relational data by several other studies as relational data can often be modeled by a graph [9] [3] [1] [13] [12].

Although the base model would be graph-structured data in our framework, our argument is that both the concept of *minimal subtree* and 'Steiner tree' would not be sufficient to handle a keyword query over several different types of data. As as result, we would adopt our own semantics that we stated in [14]. We shall elaborate on this issue in the following sections.

3. Requirements

In this section, we investigate a range of issues that need to be dealt with in order to develop a general query model for computing suitable retrieval units to answer a keyword query over heterogenous data. Solutions to these issues would form the fundamental basis for our work in future.

3. 1 Unified logical view of heterogeneous data

Any application dealing with a large collection of heterogeneous data face a daunting task of managing and querying them simply because the types of data are not limited to plain, unstructured text files or structured data that can be easily fit into a conventional Database Management Systems (DBMS). For example, a personal desktop may typically contain an extremely heterogeneous collection of data including text, video, pictures, music, emails, XML, LaTeX and Microsoft Office documents scattered across a hierarchy of folders.

The nature of relationship among *information unit*s defined in data may vary depending upon the kind of data. For example, while this relationship may be as simple as structural in document-centric XML data, both structural and semantic relationships can be considered in the case of data-centric XML. More complex one such as spatial relationship may be present predominantly in images and in the case of video data, multiple relationships such as temporal and spatial and even hierarchical relationships may be present. Similar relationships can be considered in relational data based on foreign keys and html data based on hyperlinks.

A graph-structured data model is an obvious choice for representing complex nature of heterogeneous data [6]. Similar to the approach taken by [6], our goal is to map each and every *information unit* of our data, regardless of their structural and semantic differences, as a distinct node of this graph. The basic idea is to acquire a unified logical view of physical data while maintaining a clear distinction between logical and physical representation of data.

Logical modeling of heterogeneous data based on their structural, semantic, temporal, spatial information has an important significance from a database point of view. The major objective of this approach is also to offer database-like support for heterogeneous data management so that some kind of logical data independence can be achieved. For example, a LaTeX file may have several physical representations. Several sections of its contents can either be stored either in a single file or can be stored across multiple files (one file for

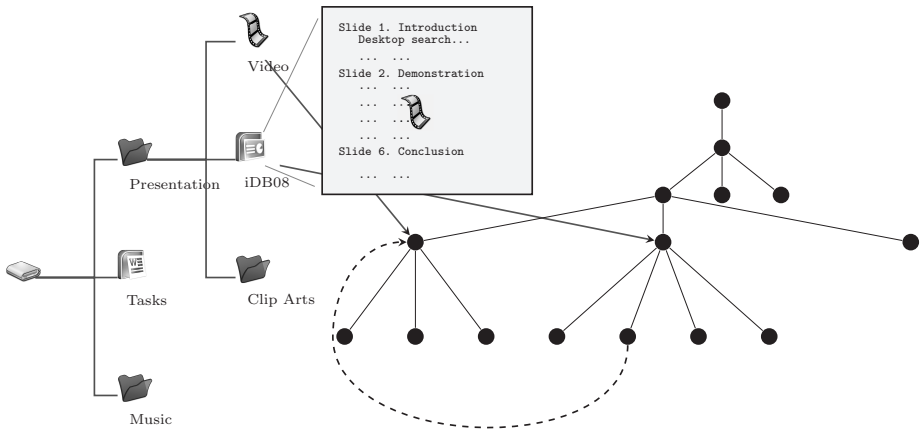

図 3 Heterogeneous data representing a personal desktop and its mapping to a graph-structured data model

each section). However, its logical representation is unique independent of its physical representation. Therefore, from the database management point of view, it is rather easier to focus on the logical representation of data. Logical data independence has played an important role in the success of Relational Database Management Systems [19]. As one can expect that the volume of data to be handled will continue to grow, this kind of database support for heterogeneous data management will be inevitable in the near future [6].

3. 2 Seamless search (uniform query semantics)

Another important issue here is that users should be able to query heterogeneous data seamlessly. What it means is that regardless of the types of data that is stored, users should be able to formulate a query uniformly and still acquire the expected result. Note that it is not unusual to have following types of answer in our framework.

- an *information unit* of a particular type
- a *retrieval unit* consisting of a set of homogeneous *information unit*
- a *retrieval unit* consisting of a set of heterogeneous *information unit*

In order to achieve this requirement, we must have a uniform query semantics. Moreover, this semantics should produce a *retrieval unit* that should be good enough no matter what set of *information unit*s this *retrieval unit* is composed of. For example, suppose query keywords are scattered across a folder, one slide of a powerpoint, a scene (consecutive shots) of a video (refer to Fig. 3), a *retrieval unit* may then consist of several of these *information unit*s that are not necessarily of the same data type. In order to achieve this kind of result, we need a uniform query semantics that should be amiable to all different kinds of data.

3. 3 Set-based approach and query optimization

Set-based approach is known to be robust and in most cases, efficient as there may be opportunities for optimization. However, there is another reason why this approach should be preferred over other ad hoc approaches. Before discussing further about the significance of set-based approach, we must look into the issue of determining a suitable *retrieval unit* in our case.

As stated earlier, the semantics of a suitable *retrieval unit* depends upon the nature of data to be retrieved. Up until now, several different semantics have been defined in the context of several different data types (refer to Section 2.). In a unified framework, we are dealing with several different kinds of data. Moreover, the framework should be flexible enough to accommodate any new types of data that are unknown yet. In our case, the semantics of a suitable *retrieval unit* defined for a keyword query over document-centric XML data in [14] is good enough. According to this semantics, a suitable *retrieval unit* is intuitively a subtree composed of at least one node (there may be more) containing each query keyword (refer to Fig. 4). Note that this semantics is different from minimal subtree or a Steiner tree as defined in other literature.

The basic idea of this semantics is to compute all potential *retrieval unit*s from a given set of *information unit*s. By doing so, naturally, there would be suitable *retrieval unit*s present in this large answer set. The issue is how to compute them efficiently. This is where set-based approach comes handy. Of course, there are other approaches such as navigational ones that are able to exploit specifically designed

index for efficient computation. However, these approaches lack extendibility and are successful only for specific cases. Set-based approach, on the other hand, may provide flexibility and extendibility in the way optimization is carried out.

There are other important issues such as the issue of efficient implementations, the issue of optimizations at the physical levels etc. One of the disadvantages of algebraic approach over algorithmic approach is immediate unavailability of means to implement operations. Here, we do not discuss further as a greal deal of work would be necessary in this area.

4. Hopes and Challenges

In the previous section, we described the major requirements for developing a general query model for keyword queries over heterogeneous data. In this section, we discuss the challenges ahead of us in order to realize such a model.

4.1 Uniform mapping of relationships among *information unit*s

As stated in Section 1., a *retrieval unit* to a keyword query needs to be computed with the help of *additional information*. This *additional information* is provided by various types of relationships such as temporal, structural, semantic etc. depending upon the nature of the data of interest. Therefore, the challenge here is not only to define a unified data model that can represent each *information unit* identified in all kinds of data but also to be able to map any type of relationship uniformly so that these relationships can be exploited for generating suitable *retrieval unit*s. One thing we are not sure yet is whether or not a simple graph-structured data is enough for mapping both hierarchical and temporal relationship in linear video data.

4.2 Defining Operations

In most cases of keyword query over XML data, a *join* operation is essential in order to compute a potential *retrieval unit*. This operation is more popularly known as the operation that finds out the *lca* (least common ancestor) of two nodes in a tree-structured data [11] [17] [21]. A broader definition is found in [14], in which this operation is termed as a *fragment join*. The basic idea is to compute a larger unit consisting of several inter-related *information unit*s. For example, according to the definition given in [14], the leftmost subtree rooted at `author` in Fig. 1 can be generated by taking the *fragment join* of the leftmost two leaf nodes. In this case, the input structure is a tree which makes the operation look relatively easy. However, when the input structure is a graph, this operation needs to be redefined as one node may have multiple parents. In this case, the ordering of the nodes based on the topological ordering of the base document, cannot be exploited to identify the relative position of two nodes. This might make the join definition more complex and as a result its implementation might become even harder. However, it is our hope that by careful implementation of the join operation, we shall still be able to take advantages of natural orderings of several types of data such as XML, linear video data, files and folders etc.

4.3 Computational Complexity

Operations on graph-structured data are notorious as computational cost can quickly grow exponentially, thus making them practically infeasible. Moreover, the nature of the semantics of *retrieval unit* in our framework, which we described in the previous section, makes the *join* operation (no matter what its definition would be) be performed a numerous number of times. Our first challenge is to estimate the required number of join operation for computing a *retrieval unit* and investigate if this can be bound by a fixed number of iterations. The basic idea is to avoid exhaustive execution of the operation. Similar kind of issue in the case of tree-structured data was described in [14].

Often, algebraic operations offer opportunity for query optimization if their equivalent expressions, which narrow down the search space, can be derived. An excellent case is described in [14], in which a special type of filters are defined for achieving logical optimization. Our challenge would be to adapt these ideas that have been proved successful for the case of tree-structured data for our new graph-structured scenario.

5. Conclusions and Future Work

Due to its simplicity, the popularity of keyword queries cannot be denied. At the same time, modern applications such as Personal Information Management Systems (PIMS) demand several different kinds of data be stored and queried in a single unified framework. In this paper, we presented some insights for deriving a general query model for keyword queries over heterogeneous data. The main characteristics of this model would be:

- Flexible enough to accommodate several different nature of data in a single unified framework.
- Unlike several other algorithmic approaches, purely algebraic, thus robust and extensible.
- Expect to acquire good *retrieval unit*s due to a broader and more meaningful query semantics.
- High hope of optimization done at the logical level.

Finding solutions to the issues that we described in Section 4. would be our immediate future work.

謝　　辞

This project has been partially supported by Grant-in-Aid

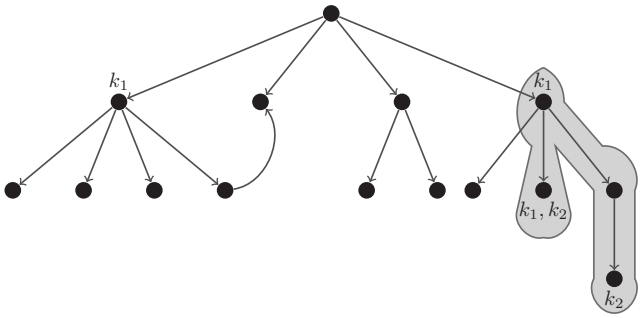

図 4 A potential retrieval unit for a keyword query {k1,k2} over a graph-structured data according to the *beyond minimal subtree* query semantics

for Scientific Research (C) (20500107).

文　献

[1] Sanjay Agrawal, Surajit Chaudhuri, and Gautam Das. Dbxplorer: A system for keyword-based search over relational databases. In *ICDE*, pages 5–16, 2002.

[2] Shurug Al-Khalifa, Cong Yu, and H. V. Jagadish. Querying structured text in an XML database. In *SIGMOD 2003*, pages 4–15, 2003.

[3] G. Bhalotia, C. Nakhe, A. Hulgeri, S. Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using BANKS. In *ICDE*, pages 431–440, 2002.

[4] S. Cohen, Y. Kanza, and B. Kimelfeld. Interconnection semantics for keyword search in XML. In *Proc. of CIKM*, pages 389–396, 2005.

[5] S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv. XSEarch: A semantic search engine for XML. In *Proc. of 29th VLDB*, pages 45–56, 2003.

[6] Jens-Peter Dittrich and Marcos Antonio Vaz Salles. iDM: a unified and versatile data model for personal dataspace management. In *VLDB*, pages 367–378, 2006.

[7] D. Florescu, D. Kossman, and I. Manolescu. Integrating keyword search into XML query processing. In *International World Wide Web Conference*, pages 119–135, 2000.

[8] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram. XRank: ranked keyword search over XML documents. In *SIGMOD*, pages 16–27. ACM, June 2003.

[9] Vagelis Hristidis and Yannis Papakonstantinou. DISCOVER: Keyword search in relational databases. In *VLDB*, pages 670–681, 2002.

[10] W.S. Li, K. S. Candan, Q. Vu, and D. Agrawal. Retrieving and organizing web pages by 'Information Unit'. In *Tenth International WWW Conference, Hong Kong, China*, pages 230–244, 2001.

[11] Y. Li, C. Yu, and H. V. Jagadish. Schema-free XQuery. In *Proc. of 30th VLDB*, pages 72–83, 2004.

[12] Fang Liu, Clement T. Yu, Weiyi Meng, and Abdur Chowdhury. Effective keyword search in relational databases. In *SIGMOD Conference*, pages 563–574, 2006.

[13] Yi Luo, Xuemin Lin, Wei Wang, and Xiaofang Zhou. Spark: top-k keyword query in relational databases. In *SIGMOD Conference*, pages 115–126, 2007.

[14] Sujeet Pradhan. An algebraic query model for effective and efficient retrieval of XML fragments. In *VLDB*, pages 295–306, 2006.

[15] Sujeet Pradhan. Towards a novel desktop search technique. In *DEXA 2007*, pages 192–201, 2007.

[16] Sujeet Pradhan, Keishi Tajima, and Katsumi Tanaka. A query model to synthesize answer intervals from indexed video units. *IEEE Trans. on Knowledge and Data Engineering*, 13(5):824–838, 2001.

[17] A. Schmidt, M. Kersten, and M. Windhouwer. Querying XML documents made easy: Nearest concept queries. In *ICDE*, pages 321–329, 2001.

[18] A. Theobald and G. Weikum. The index-based XXL search engine for querying XML data with relevance ranking. In *EDBT 2002: 8th International Conference on Extending Database Technology*, pages 477–495. Springer-Verlag, 2002.

[19] J. D. Ullman. *Principles of Database and Knowledge-Base Systems Vol. II*. Computer Science Press, 1989.

[20] Ramakrishna Varadarajan, Vagelis Hristidis, and Tao Li. Beyond single-page web search results. *IEEE Trans. Knowl. Data Eng.*, 20(3):411–424, 2008.

[21] Y. Xu and Y. Papakonstantinou. Efficient keyword search for smallest LCAs in XML databases. In *SIGMOD*, pages 527–538. ACM, June 2005.