Qualitative and Quantitative Methods in Libraries (QQML) 12,3:345-353, 2023

Conceptual data retrieval from FDB Databases

[1]Evangelia Petraki, [2]Chrysostomos Kapetis, [3]Emmanuel J. Yannakoudakis

Athens University of Economics and Business, Department of Informatics, Athens 10434, Greece

Abstract: FDB is a set theoretical model which allows the definition of multilingual databases and thesauri through a universal schema. One or more multilingual thesauri can be defined in the FDB model while the linking of each frame object (data record in terms of a traditional database) with the underlying thesauri can be implemented automatically. FDB offers administration utilities at both data and interface level, the definition of variable length objects, authority control etc. The purpose of this paper is to present the implementation of conceptual searching in any FDB database by using the information provided by one or more multilingual thesauri that have been already defined in the FDB model. Many different parameters can define the conceptual searching process in an FDB database. In this paper we firstly present briefly the FDB model, and proceed to present a) the search algorithms that exploit the information provided by the multilingual thesauri and implement conceptual searching in any FDB database, b) all the parameters that the user can define in order to determine the different search criteria.
Keywords: : databases, conceptual search, multilingual thesaurus, information retrieval, FDB

1. Introduction

This paper proposes an innovative algorithm which implements conceptual search in any FDB database. The FDB model allows the administration of any multilingual database at both data and interface level. It also supports the administration of one or more multilingual or monolingual thesauri in the same environment. In this paper we present how we can exploit the information provided from any thesaurus in the FDB model in order to apply conceptual data retrieval form the FDB databases. Moreover, the paper discusses the problem of data retrieval, research concepts and related work and proposes ways for data retrieval in the FDB model. Finally, the paper discusses future research.

In a traditional Database Management System, the data retrieval process has many limitations. In particular, the user specifies the keywords to search the

Received: 24.4.2023 Accepted: 18.9.2023 ISSN 2241-1925

database, forming in effect the search criteria by using the appropriate logical operators and expressions. The data records retrieved contain all or some of the keywords defined by the user. The traditional way of data retrieval from a database uses only the *a posteriori* relations between terms which are defined by the user and are used to express the subject of a document or record. If we take into consideration that the user does not know the contents of the database and it is impossible to include all related words (i.e. the search criteria) that may exist in the database, we realize that the information that is ignored and is not retrieved from the database may be important. Also, in cases of multilingual databases we understand that it is even more difficult for the user to include in his search criteria all the keywords in all the languages that may exist in the databases. The problem is thus evident.

Query languages like SQL are used for data retrieval in a traditional database. However, query languages are not designed to provide conceptual retrieval. Conceptual relationships between terms are defined by the *a priori* relationships and exist independently of any document and are generally recognized. As it is clear, the data retrieval process that uses both *a priori* and *a posteriori* relationships increases the number of data records retrieved and improves the effectiveness of search as it combines both of these different dimensions. A thesaurus provides *a priori* relationships between terms and can be monolingual or multilingual, while the FDB model can host one or more thesauri within the same universal schema.

A multilingual thesaurus is a set of controlled terms derived from more than one natural languages. Moreover, it identifies the correlations between the terms and the equivalent terms in all the thesaurus languages. A multilingual thesaurus allows the definition of conceptual relationships and conceptual equivalences between terms from different languages. In this paper we present how we can extend the user's queries with terms derived from one or more monolingual or multilingual thesauri in order to apply conceptual searches in any FDB database.

2. Data retrieval from databases and query languages

Different ways of data retrieval are used under different types of databases (relational, object oriented etc.). In a relational database, SQL (Structured Query Language) is frequently used for data retrieval. SQL statements are English words with a special meaning and are related to a specific task in the database. For example, the SELECT statement retrieves data from the database. The user can easily create an SQL query by using Boolean operators and the syntax of the SELECT statement but it is necessary to know the structure of the database, the names of the tables and columns and how they are correlated.

In an Object Oriented Database, Object Query Language (OQL) is used to for data retrieval. The syntax of OQL is very similar to SQL and allows the definition of different search criteria based on the keywords the user identifies. It also allows the usage of aggregate functions, group functions, ordering the results and the use of set operators like union, intersection etc. Neither of these languages allows query construction for conceptual search from databases.

Different methodologies and mechanisms have been proposed for XML databases. Yu Xu and Yannis Papakonstantinou (2005) have proposed a search mechanism and appropriate algorithms for XML documents based on keywords. Searching with keywords is highly effective for both HTML and XML documents and the proposed approach returns the smallest tree with all the keywords which form the search criteria.

In temporal databases the information stored in the database is related to temporal data. Shashi K. Gadia (1998) described temporal databases and how time values are related to data values and how validity periods of the data are defined. The TSQL/2 is a query language for such databases.

All aforementioned query languages use keywords for data retrieval and they all expect the user to have good knowledge of the database structure prior to forming a search request. The problem is that although the records that contain the given keywords are retrieved there is always the danger that some equally important records are not retrieved simply because they do not contain the necessary keywords. In other words, all retrieval evolves around simple string matching rather than truly conceptual retrieval.

Á. F. Zazo, C. G. Figuerola, J. L. A. Berrocal and E. Rodríguez (2005) claim that the major issue in information retrieval is how the user forms the queries and the key problem is how to identify all the keywords and terms that are conceptually correlated with the keywords given by the user as search criteria. Our approach introduces an algorithm which uses all the information provided by any FDB thesauri to implement data retrieval, that is, conceptual searches in any FDB database. In our approach the user identifies the keywords that will form the initial data query. Then the query is extended with terms derived from the FDB thesauri which are then used for data retrieval from the FDB database.

In section 3 the FDB model is briefly presented, in section 4 the search algorithm is illustrated and section 5 outlines the advantages of the proposed approach.

3. The FDB Model

FDB is an integrated set-theoretic model for database systems that forms a framework for defining a structure that eliminates completely the need for reorganization at the logical level, E. J. Yannakoudakis, P. K. Andrikopoulos (2007), Yannakoudakis E.J., Tsionos C.X. and Kapetis C.A (1999). FDB provides a universal model which allows the definition of any database by specifying the appropriate metadata without requiring any changes to the underlying schema. FDB also provides the definition of any multilingual or monolingual thesauri. Amongst other utilities, FDB allows administration of multilingual databases at both levels of data and interface, definition of variable length objects (records in the traditional sense), etc. Any changes that may be necessary at the data level do not affect the universal database schema but simply the identification of the appropriate metadata. The basis for the creation of the unified schema is the definition and manipulation of metadata that compose the whole structure, E. J. Yannakoudakis (1987). Accessing the information from an FDB schema becomes very easy with the use of simple

statements provided by the Conceptual Universal Database Language (CUDL), Yannakoudakis E. J., and Nitsiou M. (2006). The FDB universal schema can be used to define one or more multilingual thesauri and provides, besides traditional keyword search, conceptual searches through one or more multilingual thesauri, E. Petraki, C. Kapetis, E. J. Yannakoudakis (2013). Different algorithms have been proposed which implement the linking of each frame object with the underlying thesaurus terms automatically, enrich an existing thesaurus with terms derived from the data base, and create the core for a new thesaurus with terms derived from an FDB database, E. Petraki, C. Kapetis, E. J. Yannakoudakis (2014).

The basic elements of the model are based on the mathematical theory of unordered sets and consist of the following sets: a) entities: the unordered set of registered entities that participate in the logical schema, b) tags: the set of attributes describing each entity, c) subtags: the set of simple atomic attributes which constitute existing complex tags, d) domains: the set of all data domains, e) languages, vocabulary, messages: sets of strings or coded values that present human languages and corresponding messages, E. J. Yannakoudakis (1987). (See reference [1] for a short example).

4. Algorithm for conceptual data retrieval from FDB Databases

In what follows we present an algorithm which exploits all the information provided by the thesauri and applies conceptual searches in any FDB database. The user can define several different search parameters, as for example:

a) opt for searches with the traditional way by typing specific keywords as search criteria, without the use of the thesaurus (useThes parameter is set to FALSE).

b) opt for conceptual searches using the underlying thesauri. (useThes parameter is set to TRUE). In this case, the user may also select one thesaurus (*t*) or a set of thesauri *T* to be used during the search.

c) opt for one (*l*) or more languages (*L*) to be used in the data retrieval.

d) opt for the types of thesaurus relationships (e.g. the use of narrow terms, broader terms, related terms and synonyms, etc.). Also, the user may opt for opposite terms provided by an FDB thesaurus in order to eliminate the corresponding result set.

How the algorithm works: Two different sets are used in the algorithm, the *Opposite_terms* and the *All_keywords* set. At the beginning of the algorithm both sets are empty. Firstly, the algorithm puts all the keywords the user defines in the *All_keywords* set. Then it adds the synonyms, the broader and narrower terms and the related terms in the same set according to the user's options. Similarly, the algorithm puts all opposite terms in the appropriate set. In the second part, the algorithm checks all the data frame objects to determine whether one or more terms of the set *All_keywords* exist in the *F* set (the set that contains the words of all important data tags). If one or more terms exist then the specific data frame object will be in the result set.

We use the example from reference [1] for a more detailed explanation of the algorithm. In this example, the FDB database is a bibliographic database and the most important tags are the abstract, the title and the keywords of a book or article (tags 601, 602 and 603).

Input for the algorithm: a) The *frame_entity_numbers* and the *tag* numbers of all important entities and corresponding tags for the **data**, the **thesaurus**, the **relations** between terms and the linking between data and thesaurus terms. In the example of reference [1], a bibliographic database is stored in the FDB model. The tags which store the title, the abstract and the keywords of an article or a book are used as input data to form the *F* set. In the second part of the algorithm, all the keywords are searched in the *F* set. The algorithm presented below accepts the following input:

Frame entity numbers: **DE** (Data Entity) = 100 (book/article), **TE** (Thesaurus Entity) = 1 (thesaurus terms), **RE** (Relation Entity) = 2 (describes the relationship types between thesaurus terms), **DFO** = Data Frame Object, **TFO** = Thesaurus Terms Frame Object, **RON_NT** = Relation Object Number of the "narrower term" relationship type, **RON_BT** = Relation Object Number of the "broader term" relationship type, **RON_SYN** = Relation Object Number of the "used for" relationship type, **RON_REL** = Relation Object Number of the "related term" relationship type, **RON_OP** = Relation Object Number of the "opposite term" relationship type.

Tags: **DTAGS** (Data Tags) = 601 (abstract), 602 (title) and 603 (keywords) tags of the 100 (book/article) entity, **LTAG** (Linking Tag) = 650 of the 100 entity. LTAG is used to correlate each data frame object with the appropriate thesaurus term(s). **TTAG** (Thesaurus Term Tag) = 200 (term tag) of the 1 (thesaurus terms) entity.

Conceptual_Search Algorithm

```
Opposite_terms= {}
All_Keywords = {}
For each k ∈ K     //for each keyword defined by the user
        All_Keywords = All_Keywords ∪ {k}
        If useThes = TRUE then    // if the user wants to use thesaurus
        For each t in T loop //for each thesaurus the user has chosen
                Find_Keyword_In_Thes (t, k, exists)
                If exists = TRUE and useSYN=TRUE then
                        For each s in Synonyms_of(t,k,S,L,RON_SYN)
                        //returns all the synonyms and related terms S of the
                        keyword k
                                All_Keywords = All_Keywords ∪ {s}
                        End_Loop
                End_if
                If exists = TRUE and useNT=TRUE then
                        For each nt in NT_of(t,k,N, L,RON_NT)
                        //returns all the narrower terms of the keyword k
                                All_Keywords = All_Keywords ∪ {nt}
```

```
                    End_Loop
                End_if
                If exists = TRUE and useBT=TRUE then
                    For each bt in BT_of(t,k,B, L, RON_BT)
                     //returns all the broader terms of the keyword k
                        All_Keywords =  All_Keywords ∪ {bt}
                    End_Loop
                End_if
                If exists = TRUE and useOP=TRUE then
                    For each op in OP_of(t,k,O,L,RON_OP)
                    //returns all the opposite terms of the keyword k
                        Opposite_terms=  Opposite_terms  ∪ {op}
                    End_Loop
                End_if
                If exists = TRUE and useREL=TRUE then
                    For each rel in REL_of(t,k,R,L,RON_SYN)
                     //returns all related terms R of the keyword k
                        All_Keywords =  All_Keywords ∪ {rel}
                    End_Loop
                End_if
                End Loop //for each t in T
        End_if
// up to this point, all the keywords that are involved in the search process have
// been defined. The opposite terms have also been defined.
Result_set = {}
For each d in DFO // for each data record (data frame object)
        F = ‘’  // the F set will hold the text from all the specified data tags for
        //which we want to apply the conceptual search, abstract, title etc. in
        //our example
        For each tag in DTAGS
            F = F +’ ‘+tag
        End loop
        For each w ∈ All_Keywords Loop
            If ( w in F) OR ( w in LTAG) then // if the keyword exists in
            //F or if it is correlated with the specific data frame object
                Result_set = Result_set  ∪ {d} //data frame object
            End_If
        End_Loop
        If useOp=TRUE then
        For each w’ ∈ Opposite_terms Loop
            If w’ in F then
                    Result_set = Result_set  - {d} // this step
                    //removes all the data frame objects that
                    //include opposite terms
            End_If
        End_Loop
```

End_If
End_Loop
End Loop *//for each keyword*
End Conceptual_Search

In what follows we present a short description of the functions used in the conceptual search algorithm:

Find_Keyword_In_Thes (*t, k, exists*): This function searches the keyword *k* in the thesaurus terms; if *k* exists then it returns true, else it returns false. With a positive result this function continues to find synonyms, narrower and broader terms using the functions that follow.

synonyms_of(*t,k,S,L,RON_SYN*): The input to this function is the thesaurus *t*, the keyword *k*, the set of the languages *L*, and the relation number which corresponds to the "used for" relationship (*RON_SYN*). The function returns the set *S* of all the synonyms of keyword *k*.

NT_of(*t,k,N,L,RON_NT*): The input to this function is the thesaurus term *t*, the keyword *k*, the set of the languages *L*, and the relation number for the narrow term relation type (*RON_NT*). It returns the set *N* of all narrower terms of keyword *k*.

BT_of(*t,k,B, L, RON_BT*): The input to this function is the thesaurus *t*, the keyword *k*, the set of the languages *L*, and the relation number for the broader term relation type (*RON_BT*). It returns the set *B* of all the broader terms of keyword *k*.

OP_of(*t,k,O,L,RON_OP*) : The input to this function is the thesaurus *t*, the keyword *k*, the set of the languages *L*, and the relation number for the opposite term relationship. The function returns the set *O* of all the opposite terms of keywords *k*.

REL_of(*t,k,R,L,RON_SYN*): The input to this function is the thesaurus *t*, the keyword *k*, the set of the languages *L*, and the relation number which corresponds to the "relative term" relationship (*RON_REL*). The function returns the set *R* of all the synonyms of keyword *k*.

The above version of the conceptual search algorithm that can be applied to any FDB Database is a brief and concise presentation that exploits all the features offered by using one or more thesauri for data retrieval. Many variants of this algorithm can be defined; the algorithm presented is better because it is simple and comprehensive.

5. Advantages and issues for further research

FDB is an integrated database management system which provides a universal schema that allows the definition of any multilingual database and thesauri by enabling the setting of appropriate metadata. It allows data retrieval by using both free text techniques and conceptual searches through the use of multilingual thesauri.

In a traditional database system, data retrieval is carried out by using the query languages like SQL, OQL etc. The data queries are formed with the

keywords the user defines. This approach expects the user to have full knowledge of the structure of the database, while it uses only the *a posteriori* relations between terms. The *a priori* relations between terms are ignored by these approaches. The *a priori* relationships exist independently of any document and can be expressed by a thesaurus hierarchy. The research presented here facilitates the use of any monolingual or multilingual thesaurus for data retrieval from any FDB database. Each query can be extended with terms (related terms, synonyms, narrower terms, etc.) derived from one or more thesauri. This approach is very useful especially for large databases as it provides conceptual data retrieval from several databases. Another advantage of the proposed method is that the user can run the same algorithm for different thesauri and databases. This offers significant flexibility because each query can be created by using terms from different thesauri. In the FDB model databases and thesauri are defined within the same unified and integrated system. A universal schema can host any database and several thesauri.

The algorithm presented here can be expanded and improved by including a weighting method in order to avoid query extension with wrong terms. With such improvements it will be possible to increase recall and precision in data retrieval.

References

[1] E. Petraki, C. Kapetis, E.J. Yannakoudakis, (2013). Conceptual database retrieval through multilingual thesauri, *Computer Science and Information Technology* 1(1): 19-32,

[2] E. J. Yannakoudakis, P. K. Andrikopoulos, (2007). A set-theoretic data model for evolving database environments, In Proceedings of the *International Conference on Information & Knowledge Engineering*, IKE 2007, Las Vegas, Nevada, USA.

[3] Yannakoudakis E.J., Tsionos C.X. and Kapetis C.A, (1999). A new framework for dynamically evolving database environments, *Journal of Documentation*, Vol. 55, No. 2, pp. 144-158.

[4] E. J. Yannakoudakis, (1987). An efficient file structure for specialised dictionaries and other 'lumpy' data, *International Journal of Information Processing & Management*, Vol. 23, No. 6, pp. 563-571.

[5] E. Petraki, C. Kapetis, E.J. Yannakoudakis, (2014). Automated thesaurus population and management, In *6th International Conference on Qualitative and Quantitative Methods in Libraries*, Istanbul

[6] Yu Xu, Yannis Papakonstantinou, (2005). Efficient Keyword Search for Smallest LCAs in XML Databases, DOI: 10.1145/1066157.1066217 Conference: Proceedings of the *ACM SIGMOD International Conference on Management of Data*, Baltimore, Maryland, USA, June 14-16, 2005

[7] Shashi K. Gadia, (1988). A Homogeneous Relational Model and Query Languages for Temporal Databases, *ACM Transactions on Database Systems*, Vol. 13, No. 4, December 1988, Pages 418-448.

[8] Yannakoudakis E. J., and Nitsiou M. (2006), A new conceptual universal database language (CUDL), In *2nd International Conference From Scientific Computing to Computational Engineering*, Athens, Greece.

[9] Ángel F. Zazo, Carlos G. Figuerola, José L. Alonso Berrocal, Emilio Rodríguez, (2005), Reformulation of queries using similarity thesauri, *Information Processing & Management*, Volume 41, Issue 5, September 2005, Pages 1163–1173

[10] R Baeza-Yates, B Ribeiro-Neto, (1999). *Modern Information Retrieval*, Book, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, ISBN:020139829X, ACM Press