Unstructured Data Warehouse Processing System Based on an Uniform Set of Functions

Anton Mikhailyuk, Denis Zamiatin, Andrey Petrashenko

Applied Mathematic Department, National Technical University of Ukraine "Kyiv Polytechnic Institute", Peremohy Ave., 37, Kyiv, UKRAINE, E-mail: petrashenko@gmail.com

Abstract – we propose to provide unstructured data warehouse as a set of entities described by a set of attribute values for a given set of warehouse attributes. A set of basic functions is defined for the warehouse, which is represented as functions that return a new warehouse. Union, complement, intersection, filtering, grouping etc can be identified as the functions. Warehouses or predicates could be defined as the arguments of functions for the attribute values. The filtering using this predicate will be the implementation of full-text search. The proposed approach allows formalizing business processes for processing unstructured data warehouses as a sequence of basic functions, and optimization of business processes based on formal transformation rules of these functions.

Key words – unstructured data warehouse, full-text search, entity-attribute-value.

I. Introduction

The modern global information space is characterized by large volumes of unstructured text information. No formal allocation of attributes with a steady increase in volumes of text resources greatly complicates the cataloguing procedure and information retrieval. The problem is aggravated because many subject areas there are specific algorithms for structuring textual information that leads to the need of specialized software. At the same time, today there is no unified approach to requirements analysis, design and development of software processing unstructured text resources targeted for use in a particular subject area. In this paper a generalized approach to formal representation of algorithms for processing unstructured text information is shown.

II. Model of unstructured text warehouse

To concretize the problem, assume that in a considered text warehouse we can isolate specific text fragments, which later will be called documents. Thus a warehouse W consists of documents set D_i, $i = 1...n$. The property of an unstructuredness can be interpreted as a lack of documents common characteristics, such as, for example, the type of document using which they could be formally divided into certain groups. Therefore, as a source of information for structuring only attributes of a document can act, including its immediate text. Let A is the set of all possible attributes of documents $D_i \in W$, then the document D_i can submit as a tuple consisting of the attributes set $A_j \in A$ of a document and its values V_j:

$$D_i = <A_j : V_j>$$

Thus, an unstructured text warehouse can be represented as a documents set, each of which consists of a values set of certain attributes.

III. Operations on the warehouses

For the formalization of algorithms for processing data in the warehouse, it is necessary to develop a set of unified operations, the result of which should be the new warehouse. It will simplify the consistent using of operations. At this point in the set of such operations is proposed to include the basic set-theoretical actions and additional selection operation.

Among the set-theoretic operations it is appropriate to use:

a) union
$$W_1 \cup W_2 = \{D \mid D \in W_1 \vee D \in W_2\};$$

b) intersection
$$W_1 \cap W_2 = \{D \mid D \in W_1 \wedge D \in W_2\};$$

c) subtraction
$$W_1 - W_2 = \{D \mid D \in W_1 \wedge D \notin W_2\};$$

The selection operation get the value of certain characteristic Boolean function f as an argument, which is consistently applied to all documents in the warehouse, and returns a new warehouse that contains only documents for which the function is true:

$$W.where(f) = \{D \mid D \in W \wedge f(D) = True\}.$$

In addition to operations that return the warehouse as a result, to obtain information about attribute values it is appropriate to include a function that defines the set of available attribute values:

$$W.values(A_j) = \{V_j \mid \exists A_j{:}V_j \wedge D \in W \wedge A_j \in D\}.$$

IV. Operations on the attribute values

As the characteristic function f should be a Boolean function, which in some way examines the attributes values of a document. For many domains it is appropriate to use the following functions:

a) filtering function by value

$A_j{:}V_j\ \theta\ const = True$, if $A_j \in D_i \wedge V_j\ \theta\ const$, where $\theta \in \{=, \neq, >, <, \geq, \leq\}$;

b) filtering function by content

$const$ in $A_j{:}V_j = True$, if $A_j \in D_i \wedge V_j$ <contains> $const$,

where a <contains> means including *const* as substring.

Depending on the domain specific it is possible to append additional operations that take into account the relevant algorithmic features.

Suggested representation of a text warehouse with included operations allows describing formally the steps needed to apply over the resource to get a result.

V.The basic processing documents modes implementation using basic unified set of operations

Search by attributes

The problem add up to using of selection operations, i.e. if you want to find all documents where the attribute "City" equal to "Kyiv", the result will be:

$$W.where(A(\text{“City”}) = \text{“Kyiv”}).$$

Search by multiple terms will be determined by the intersection of selections by each criterion:

$$W.where(A(\text{“City”}) = \text{“Kyiv”}) \cap W.where(A(\text{“Year”}) = \text{“2009”})$$

Similarly, the search mode by value is in a range can be expressed through the intersection operation:

$$W.where(A(\text{“Year”}) > \text{“2005”}) \cap W.where(A(\text{“Year”}) < \text{“2009”})$$

Full text search

Full-text search mode returns a set of documents containing a given word. This mode can be impelemented using of the filtering function by content:

$$W.where(\text{“word” } in\ A(\text{“Text”}))$$

Widely used variations of searching for documents, which contains all the words from the set, any of them or without given words [1]. Such combinations can be expressed according to the formulas:

$$W.where(\text{“word1” } in\ A(\text{“Text”})) \cap W.where(\text{“word2” } in\ A(\text{“Text”}))$$

$$W.where(\text{“word1” } in\ A(\text{“Text”})) \cup W.where(\text{“word2” } in\ A(\text{“Text”}))$$

$$W - W.where(\text{“word” } in\ A(\text{“Text”})).$$

Cataloging

Building a catalogue of documents in general includes the solution of two tasks: to build a catalogue tree and to determine the set of documents related to the selected branch. The tree construction is add up to the determination of all the available attribute values at a certain catalogue level hierarchy. For example, documents with attribute *A*("Year") = "2009" and *A*("City") = "Kyiv" the calculation for the attribute list of branches *A*("Author") look like this:

$$(W.where(A(\text{“City”}) = \text{“Kyiv”}) \cap W.where(A(\text{“Year”}) = \text{“2009”})).values(A(\text{“Author”}))$$

The second task - finding a set of documents related to the selected branch is add up to filtering:

$$W.where(A(\text{“City”}) = \text{“ Kyiv”}) \cap W.where(A(\text{“Year”}) = \text{“2009”} \cap W.where(A(\text{“Author”}) = \text{“Shevchenko”})$$

So, it is shown the possibility of traditional text processing methods implementation using the unified set of operations.

VI. Application in the modern relational database management systems

The document warehouse model which represents data as a set of attribute tuples provides effective implementation based on entity-attribute-value approach [2,3], which allows to adapt the data warehouse to an arbitrary document structures without modifying the existing database schema. It reduces the cost of support integrity. In this case, the warehouse can be represented as two linked tables containing attributes and their values, respectively.

The proposed set of unified operations has direct analogy with the operations of relational algebra [4], which allows creating effective implementation of unstructured text processing software using the modern database management systems.

This approach is tested during the process of an information infrastructure formation in several universities and government institutions (Fig.1).

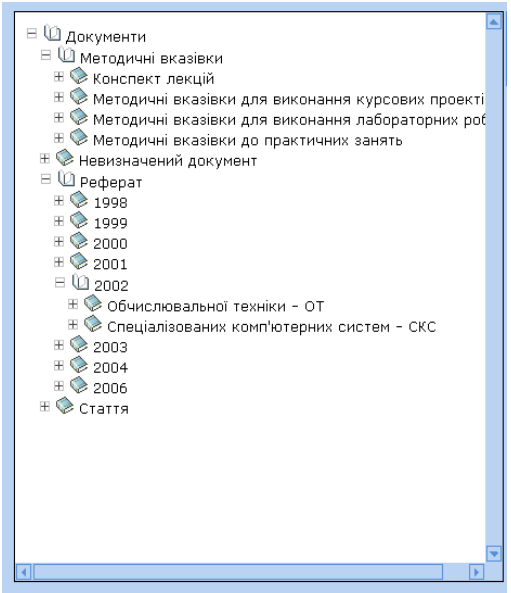

Figure 1. Example of catalogue tree construction mode.

Conclusion

The proposed approach allows describing formally the business processes for unstructured data warehouses as a set of unified operations and functions and creating their effective implementation using the modern database management systems.

Thus, the described set of functions that can be the base for creating flexible and scalable information-analytical systems targeted for processing unstructured text warehouses.

References

[1] Замятін Д.С., Михайлюк В.А., Петрашенко А.В. Підвищення продуктивності повнотекстового пошуку шляхом реорганізації подання інвертованих індексів // Науковий вісник чернівецького університету. Збірник наукових праць. Випуск 426. — Чернівці. — 2008. — С. 63—67.

[2] Jennings, Roger (2009), "Retire your Data Center", Visual Studio Magazine Feb 2009: 14-25

[3] McDonald, C.J.; Blevins, L.; Tierney, W.M.; Martin, D.K. (1988), "The Regenstrief Medical Records", MD Computing (5(5)): 34-47

[4] К.Дж.Дейт Введение в системы баз данных, 8-е издание. – К.; М.; СПб.: Издательский дом «Вильямс», 1999. – 1328 с.