THE METHOD OF UNSTRUCTURED TEXT-BASED WAREHOUSES SYSTEMATIZATION BASED ON COMPOSITE ATTRIBUTES

A.Mykhailyuk. Boris Grinchenko Kyiv University
A.Petrashenko, D.Zamiatin. National Technical University of Ukraine "KPI"

The modern global information space is characterized by large volumes of unstructured electronic resources, which are presented in the form of text documents, emails, web-pages, etc. Systematization of such resources is complicated by the lack of a uniform presentation of common attributes, including personal information, dates and their ranges, names, toponyms, etc. For example, several articles written by one author may be signed by different ways: "Taras Shevchenko, T. Shevchenko, Shevchenko T." Existing information retrieval tools are not able to associate such variants with a single author.

Modern methods of text analysis and cataloguing of resources based on the theoretical apparatus of unstructured data stores, designed in particular W. Inmon and others [1, 2], use semantic oriented methods that have significant computational cost and relatively low accuracy. The approach is proposed in [3] to handling unstructured text storage based on a unified set of operations allows organizing resources by means of an automated retrieving of links between the attributes values.

Therefore, the purpose of this work is to improve the quality of organizing these resources by way of a composite attributes formal notation developing.

Consider the model of a text-oriented unstructured storage. Let the warehouse W consists of a document set D_i, $i = 1...n$, where A — set of all possible document attributes $D_i \in W$. Then the document D_i can be applied as a tuple set consisted of the document attributes set $A_j \in A$ and their values V_{ij}:

$$D_i = \{<A_j : V_{ij}>\}$$

Let the attribute $A_j \in A$ is *composite* if its value V_{ij} for the document $D_i \in W$ is able to apply as a specific function of the other attribute values of this document:

$$V_{ij} = f(V_{in}, V_{in+1},...,V_{in+k})$$

The calculation of the value V_{ij} of function f, which is predefined on the basis of the domain specific properties, will link the document with other documents D_i for which attribute A_j matches V_{ij}, even if the document D_i has no the attribute A_j. Several functions may exist for each composite attribute. This matching provides an opportunity to find hidden links between the documents received from different sources and different views on attribute values.

Depending on the data types of attributes, the function f can be formally written as a sequence of arithmetic operations, strings operations and type conversions (Table 1).

Here are some examples of the formal composite attributes notation usage to organize unstructured text resources.

Let we need to find in the warehouse all of the documents by a particular author. Different notations of the author's name are used in various publications, for example: "Surname Fn.Sn." (option 1), "Fn Surname" (option 2) and even Prizvysche I. (option 3). Someone knowledgeable in the subject area, may create some rules for matching the attribute values based on the basic attributes:

A(“Surname”) + “ ” + A(“Fn”)[0] + “.” + A(“Sn”)[0] + “.” (option 1);
A(“Fn”) + “ ” + A(“Surname”) (option 2);
translit(A(“Surname”)) + “ ” + translit(A(“Fn”)[0]) + “.” (option 3),

where *[n]* — an substring operation (n-th symbol) and translit () – the transliteration function. Based on these rules every option of the author's name can be transformed into a single form and matched with, for example, the author’s portfolio.

Table 1

Operations on attribute values, depending on the data types

	String	Number	Date
String	Concatenation, substring, transliteration and retransliteration	Type conversion	Type conversion using a pattern
Number	Type conversion	+, -, *, /, other operations and functions	Type conversion using a pattern
Date	Type conversion using a pattern and extracting date elements	Extracting date elements	Dates subtraction

Another example of the composite attributes usage is the determination of the event date. If the date of an article publication may be written in several formats: “DD.MM.YYYY” (option 1) and “DD month YYYY” (option 2), then to find all articles published in a given year, we can use these rules:

int(A(“Date”)[:2]) + 2000 (option 1);
int(A(“Date”)[:4]) (option 2),

where *[:m]* — *m*-characters substring operation started from the end of line and *int()* – type conversion from string to integer.

Documents association using composite attributes, such as cataloging mode, can take place with the following algorithm (Fig. 1). The first step is the document set selection for the non-composite attribute. In terms of unified operations set introduced in [3] a catalogue branch will be determined by the formula:

$$\{W.where(A' = V) \mid \forall\ V \in W.values(A)\},$$

where W — the document warehouse, A' — value of a certain attribute A, *where* — the selection operation by the value of attributes, *values* — finding the set of all available attribute values. After grouping these documents in some branches, documents is searched for the available attributes that are used in each function f_i which defines composite attributes:

$$W' = \{\cap W.where(exists(A)) \mid \forall\ A \in Arg(f_i)\},$$

where $Arg(f_i)$ — set of function f_i arguments, $exists(A)$ — function that defines the attribute A in a document. The values of attributes are calculated for the found documents. Based on these attributes values further formation of catalogue branches is done. So the set of documents that match a catalogue branch with value V will consist of:

$$W.where(A' = V) \cup \{\cup W'.where(f_i = V) \mid \forall f_i\}$$

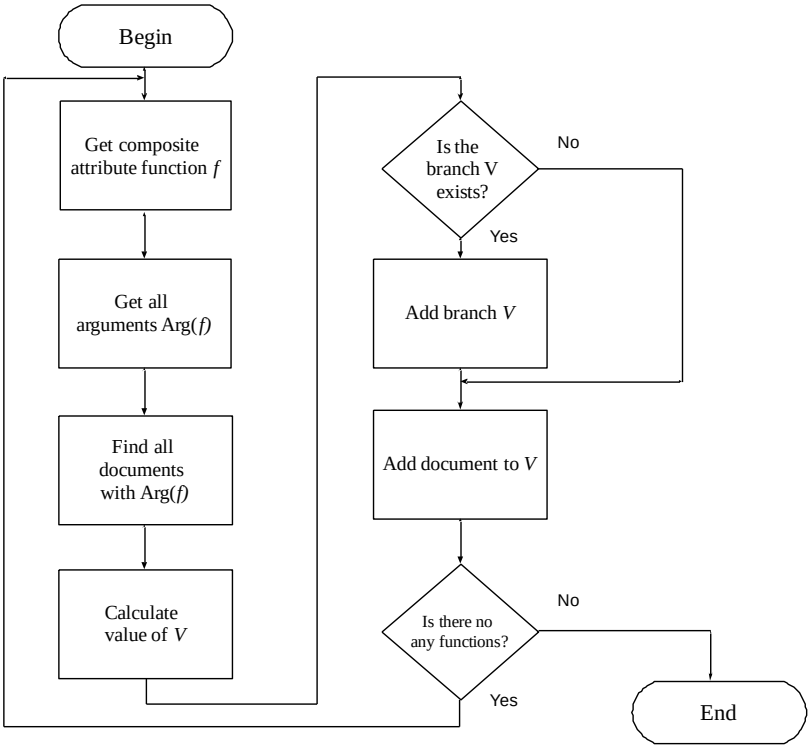

Fig. 1. Cataloging algorithm scheme.

Thus, the method of documents systematization in text-oriented unstructured warehouses is suggested. This method is based on the formal notation of composite attributes. It allows to find documents that are not directly related by attribute values and to identify hidden relations. Further research is planned to implement suggested cataloging algorithm in unstructured warehouses software for scientific and educational field.

References

1. W. H. Inmon, A. Nesavich Tapping into Unstructured Data: Integrating Unstructured Data and Textual Analytics into Business Intelligence. — Prentice Hall. — 2007. — 264 p.
2. R. Feldman, J. Sanger The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. — Cambridge University Press. — 2007. — 410 p.
3. Mykhailyuk A., Zamiatin D., Petrashenko A. Unstructured Data Warehouse Processing System Based on an Uniform Set of Functions // Proceedings of the 4-th International Conference ACSN-2009 "Advanced Computer Systems and Networks: Design and Application". — Lviv. — 2009. — P. 117-119