Using textual dimensions in Data Warehousing processes

M.J. Martín-Bautista[1] C. Molina[2] E. Tejeda[3] M.A. Vila[1]

[1] University of Granada, Spain
[2] University of Jaen, Spain
[3] University of Camagüey, Cuba

Abstract. In this work, we present a proposal of a new multidimensional model handling semantical information coming from textual data. Based of a semantical structures called AP-structures, we add new textual dimensions to our model. This new dimension aloow the user to enrich the data analisis not only using lexical information (a set or terms) but the meaning behind the textual data.

1 Introduction

The information and knowledge management is a strategic activity for the success of the companies. Textual information takes part of this information, specially from the coming of the Internet. However, it is complex to process this kind of data due to the lack of structure and its heterogeneity. For this reason, there exist not many integrated tools processing this textual information together with other processes such as Data Mining, Data Warehouse, OLAP, etc.

In particular, and as far as we know, there exists no implementations of Data Warehousing and OLAP able to analyze textual attributes in databases from a semantical point of view. The proposal in this work try to solve this problem.

This work shows a multidimensional model with semantical treatment of texts to build the data cubes. In this way, we can implement a Data Warehousing with OLAP processing using this model. That is, a process that be able to get useful information from textual data coming from external files or from textual attributes in a database.

For this purpose, this paper is organized as follows: in the next section, we review the literature related to our proposal, specially those works about Data Warehousing with texts. In section 3, we presents a classical multidimensional model as a base for the extension without textual dimensions. Section 4 collects the formal model proposed and in 5 an example is shown. The papers finishes with the main conclusions.

2 Related Work

In this section, we include some of the most relevant works about Data Warehousing related to processing of textual data. In most of them, different techniques

are used to manage textual data and to incorporate them in a multidimensional model, but the source of texts are usually XML documents or texts with some internal structure.

The creation of a Data Warehouse of Words (WoW) is proposed in [3]. This proposal extracts a list of words from plain text and XML documents and stores the result in DataCubes. The proposal in [9] is based on XML too, and propose a distribute system to build the datacubes in XML. In [10], short texts such as emails and publications are transformed into a multidimensional models and can be queried.

Obviously, the restriction of using XML or structured texts implies generally the intervention of the user to generate them and structure them. In our proposal, the entry can be either a set of external files in plain text, XML, or any other format. However, our approach also considers the textual attributes in a database; in fact, whatever the entry data, they are transformed into an attribute in a database which will be a textual dimension in the future. This transformed textual attribute has two main advantages with respect to other textual representations. First, it takes the semantic of the text. In this process althought statistic a method is applied, the resulting structure is directly understandable by the user. Second, it can be obtained automatically and without the user's intervention. The process to perform this transformation is shown in section 4.4. Due to the semantical treatment of the textual data, a semantical dimension in the data cube is generated. Data Warehousing and OLAP processes are then performed.

3 Background

3.1 The *classical* multidimensional model

The model presented here is a resume of the characteristics of the first models proposed in the literature of Data Warehousing and OLAP [1], [4], since we do not consider that there be a standard one [8]. This model is the base of most of the proposals reviewed in Section 2, and also the starting point to achieve our goal: a new multidimensional model with a more powerful textual processing.

In a classical multidimensional model we can consider the following elements:

- A set of *dimensions* $d_1, ..d_n$ defined in a database. That is, attributes with a discrete domain belonging to the database scheme. The data are grouped attending these attributes. Each dimension d_i has associated:
  - A basic domain $D_i = \{x_1....x_{m_i}\}$ of discrete values so, each tuple t of the database takes an unique and well determined value x_i in the attribute d_i. Let us note $d_i[t] = x_i$.
  - A grouping hierarchy that allow us to consider different values for the analysis. Such a hierarchy $\mathcal{H}_i = \{\mathcal{C}_{i1}...\mathcal{C}_{il}\}$ is formed by partitions D_i in a way that:

$$\forall k \in \{1, 2...l\}\ \mathcal{C}_{ik} \subseteq \mathcal{P}(D_i)\ \mathcal{C}_{ik} = \{X_{ik}^1, .., X_{ik}^h\}$$

being $\forall j, r\ X_{ik}^j \bigcap X_{ik}^r = \emptyset$ and $\bigcup_{j=1}^{h} X_{ik}^j = D_i$.
The hierarchy $\mathcal{H}_i$ is an inclusion reticulum which minimal element is D_i, considering element by element, and the maximal is D_i considering a partition of just one element.

- A numeric measure V associated to these dimensions, so we can always obtain $V = f(Y_1, Y_2..Y_n)$ where $Y_1..Y_n$ are values of the dimensions considered above. We must point out that these values may not be exactly the same as the ones in the domain, but the ones in some partition of the hierarchy. That is, if we consider the level $\mathcal{C}_{ik}$ in the dimension d_i, then $Y_i \in \mathcal{C}_{ik}$. This measure V can be:
  - A count measure which gives us the number of tuples in the database that verify $\forall i \in \{1, ..n\}\ d_i[t] \in Y_i\}$
  - Any other numerical attribute that is semantically associated to the considered dimensions.
- There exits also an aggregation criterion of V, AGG, which is applied when 'set' values are considered in any of the dimensions. That is,

$$V = f(x_1, ..Y_k, ..x_n) = AGG_{x_k \in Y_k} f(x_1, ..x_k, ..x_n)$$

AGG can be a sum, SUM, or any other statistical function like the average, AVG, the standard deviation, STD, etc. Obviously, is the measure is the count one, the aggregation function is SUM.

From the concept of data cube, the normal operations are defined. They correspond to the different possibilities of analysis on the dimensions (roll-up, drill-down, slice and dice).

We must also remark that there are other approaches in the literature where there are no explicit hierarchies defined on the dimensions, like the one in [2].

4 Formal model

Due to space limitation, in this paper we only present the main aspect needed to understand de proposal. The complete model can be found in [7, 6, 5].

4.1 AP-Set definition and properties

Definition 1. AP-Set
Let be $X = \{x_1...x_n\}$ any referential and $\mathcal{R} \subseteq \mathcal{P}(X)$ we will say $\mathcal{R}$ is and AP-Set if and only if:

1. $\forall Z \in \mathcal{R} \Rightarrow \mathcal{P}(Z) \subseteq \mathcal{R}$
2. $\exists Y \in \mathcal{R}$ *such that* :
   (a) $card(Y) = max_{Z \in \mathcal{R}}(card(Z))$ *and not exists* $Y' \in \mathcal{R}$ *such that* $card(Y') = card(Y)$
   (b) $\forall Z \in \mathcal{R}; Z \subseteq Y$

The set Y of maximal cardinal characterizes the AP-Set and it will be called *spanning set of* $\mathcal{R}$. We will denote $\mathcal{R} = g(Y)$, that is $g(Y)$ will be the AP-Set with spanning set Y.

We will call *Level of* $g(Y)$ to the cardinal of Y. Obviously, AP-Set of level equal to 1 are the elements of X, we will consider the empty set $\emptyset$ as the AP-Set of zero level.

It should be remarked that the definition 8 implies that any AP-Set $g(Y)$ is in fact the reticulum of $\mathcal{P}(Y)$

Definition 2. AP-Set Inclusion
Let be $\mathcal{R} = g(R)$ *and* $\mathcal{S} = g(S)$ *two AP-Sets with the same referential:*

$$\mathcal{R} \subseteq \mathcal{S} \Leftrightarrow R \subseteq S$$

Definition 3. Induced sub-AP-Set *Let be* $\mathcal{R} = g(R)$ *and* $Y \subseteq X$ *we will say* $\mathcal{S}$ *is the* sub-AP-Set induced by Y *iff:*

$$\mathcal{S} = g(R \bigcap Y)$$

Definition 4. Induced super-AP-Set *Let be* $\mathcal{R} = g(R)$ *and* $Y \subseteq X$ *we will say* $\mathcal{V}$ *is the* super-AP-Set induced by Y *iff:*

$$\mathcal{V} = g(R \bigcup Y)$$

4.2 AP-Structure definition and properties

Once we have established the AP-Set concept we will use it to define the information structures which appear when frequent itemsets are computed. It should be considered that such structures are obtained in a constructive way, by initially generating itemsets with cardinal equal to 1, next these ones are combined to obtain those of cardinal equal 2, and by continuing until getting itemsets of maximal cardinal, with a fixed minimal support. Therefore the final structure is that of a set of AP-Sets, which formally is defined as follows.

Definition 5. AP-Structure
Let be $X = \{x_1 ... x_n\}$ *any referential and*
$S = \{A, B, ...\} \subseteq \mathcal{P}(X)$ *such that:*

$$\forall A, B \in S\,;\; A \not\subseteq B\,,\; B \not\subseteq A$$

We will call AP-Structure of spanning S*,*
$\mathcal{T} = g(A, B, ...)$*, to the set of AP-Set whose spanning sets are* $A, B, ...$

Now we will give some definition and properties of these new structures.

Definition 6. *Let be* $\mathcal{T}_1$*,* $\mathcal{T}_2$*, two AP-Structures with the same referential:*

$$\mathcal{T}_1 \subseteq \mathcal{T}_2 \;\Leftrightarrow\; \forall \mathcal{R}\ \textit{AP-Set of}\ \mathcal{T}_1\ ,$$

$$\exists \mathcal{S}\ \textit{AP-Set of}\ \mathcal{T}_2\ \textit{such that}\ \mathcal{R} \subseteq \mathcal{S}$$

It should be remarked that the inclusion of AP-Set is that which is given in the definition 2.

Extending the definitions 3 and 4 we can defined the *Induced AP-Substructure* and *Induce AP-Superstructure* (see [6] for details).

4.3 Matching sets with AP-structures

Now we will establish the basis for querying in a database where the AP-structure appears as data type. The idea is that the users will express their requirements as sets of terms and in the database will be AP-structures as attribute values, therefore some kind of matching has to be given.

Two approaches are propposed: *weak* and *strong matching*. A detail definition can be found in [5, 6]. The idea behind the matching is compare the spanning sets for the AP-struture and the set of terms given by the user. The *strong matching* consider that the set of terms by tue user and the AP-structure match if all the terms are include in a spanning set. The *weak matching* relaxes the condition and return *true* if at least on of the term is included ina spanning set.

These matching criterias can be complemented by giving some measures or indexes which quantify these matchings. The idea is to consider that the matching of a long set of terms will have an index greater than other with less terms, additionally if some term set match with more than one spanning set will have an index greater than that of the other one which only match with one set. Obviously two matching indexes can be established, but both two have similar definitions.

Definition 7. strong(weak) matching index

Let be an AP-structure $\mathcal{T} = g(A_1, A_2, ..., A_n)$ *with referential* X *and* $Y \subseteq X$*, we define the* strong(weak) matching index between Y and $\mathcal{T}$ *as follows:*

$\forall A_i \in \{A_1, A_2, ..., A_n\}$ *we denote* $m_i(Y) = card(Y \bigcap A_i)/card(A_i)$, $S = \{i \in \{1, ..., n\} | Y \subseteq A_i\}$, $W = \{i \in \{1, ..., n\} | Y \bigcap A_i \neq \emptyset\}$.

Then we define the strong and weak matching indexes between Y and $\mathcal{T}$ *as follows:*

$$\textbf{Strong index} = S(Y|\mathcal{T}) = \sum_{i \in S} m_i(Y)/n$$

$$\textbf{Weak index} = W(Y|\mathcal{T}) = \sum_{i \in W} m_i(Y)/n$$

Obviously:

$$\forall Y \text{ and } \mathcal{T}\ ,\ S(Y|\mathcal{T}) \in [0,1]\ ,\ W(Y|\mathcal{T}) \in [0,1] \text{ and } W(Y|\mathcal{T}) \geq S(Y|\mathcal{T})$$

4.4 Transformation into an AP-attribute

In this section we briefly describe the process to transform a textual attribute in an AP-structure valuated attribute, what we call an *AP-attribute*.

1. The frequent terms associated to the textual attribute are obtained. This process includes cleaning process, empty words deleting process, synonymous management process using dictinaries, etc. Then we get a set of basic terms T to work with. In this point the value of textual attribute on each tuple t is subset of basic terms T_t. This consideration allow us to work with the tuples as in a transactional database regarding the textual attribute.
2. Maximal frequent itemsets are calculated. Been $\{A_1, .., A_n\}$ the itemsets, the AP-structure $S = g(A_1, .., A_n)$ includes all the frequent itemsets, so we can consider the AP-structure to cover the semantic of the textual attribute.
3. Once we have the global AP-structure, we obtain the AP-structure associated to tuple t: if T_t is the set of terms associated to t, the value of AP-attribute for the tuple is:
$$S_t = g(A_1, .., A_n) \bigwedge T_t$$

This process obtains the domain for any AP-attribute.

Definition 8. Domain of an AP-attribute *Considering a database to build the AP-attribute A with global structure* $(A_1, ..., A_n)$*, the domain of attribute A is*

$$D_A = \{R = g(B_1, ..B_m), /, \forall i \in \{1, .., m\}, \exists, j \in \{1, .., n\} such\ that B_i \subseteq A_j\}$$

So D_A is the set of all sub-AP-structures of the global AP-structure associated to the attribute, because these are all the possible values for attribute A according to previous constraint.

As an example let consider a simplification of data of patient in an emergencies service at an hospital. Table 1 shows some records stored in the database. Attributes *Patient number (no)*, *Waiting time*, *Town* are classical attributes. *Diagnosis* is textual attribute that stores the information given by the medical doctor for the patient.

No.	Waiting time	Town	Diagnosis
1	10	Granada	pain in left leg
2	5	Gojar	headache and vomit
3	10	Motril	voimit and headache
4	15	Granada	rigth arm fractured and vomit
5	15	Armilla	intense headache
...	...	...	...

Table 1. Example of database with a textual attribute

After applying the propposed process, we tranform the textual attribute into an AP-attribute. Figure 1 shows the AP-structure obtained for the diagnosis attribute. The sets at the top of the structure are the spanning set of the attribute. The other are all the possible subsets with the elements in the spanning sets.

Then the database is transformed to stores the spanning sets associated to each records as shown in Table 2.

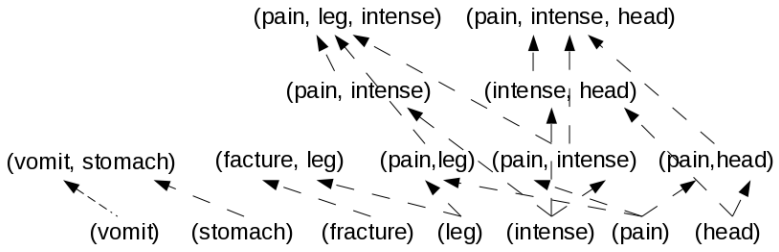

Fig. 1. Global AP-structure

4.5 Dimension associated to an AP-attribute

To use the AP-attribute on a multidimensional model we need to define a concept hierarchy and the operations over it. We need first some considerations.

- Although the internal representation of a AP-attribute are structures, the input and output for the user is carry out by means of terms sets ("sentences"), which are spanning for the AP-structures.
- This will be the same case for OLAP. The user will give as input a set of sentences, as values of the dimension, although these sentences are values of the AP-attribute domain.
- According to definition 8 we are working with a structure domain and closed when we consider the union. So, a set of elements of the domain is include in the domain. Then, the basic domain for a dimension associated to an AP-structure and the domain of the hierarchies is the same.

According to these considerations we have the following definition.

Definition 9. AP-structure partition associated to a query

Let $C = \{T_1, .., T_q\}$ where $T_i \subseteq X$ is subset of "sentences" given by an user for a dimension of a AP-attribute. Been S the global AP-structure associated to that attribute. We define the **AP-structure partition associated to** *C as:*

$$\mathcal{P} = \{S_1, .., S_q, S_{q+1}\}$$

where

$$S_i = \begin{cases} S \bigwedge T_i & \text{if } i \in \{1, .., q\} \\ S \bigwedge (X - \bigcup_{i=1}^{q} T_i) & \text{otherwise} \end{cases}$$

Now we can introduce a multidimensional model as define in section 3.1 that use an AP-dimension:

- $\forall i \in \{1, .., q\}$ $f(.., S_i, ..)$ is an aggregation (count, or other numeric aggregation) associated with the tuples that satisfy T_i in any way.
- $f(..., S_q, ...)$ is an aggregation associated to the tuples not matching any sentences in T_i, or part of them. That means, the sentences that are not related with the sentences given by the user.

Obviously, the matching concept and the considerate aggregations have to be adapted to the characteristics of an AP-dimension.

5 Example

Let consider the example introduced in Section 4.4 about an emergencies service at an hospital to show how queries are answered in a datacube with the AP-attribute.

Let suppose the partition for the following query:

$$C = \{(pain, intense), (vomit)\}$$

If we choose the count aggregation and the *weak matching* (definition **??**) the results are shown in Table 3. On the other hand, if we use the strong matching (definition 9) the results are the one collected in Table 4. As it was expected, when considering the *weak matching* more records satisfy the constraint than for the very strict *strong matching*

We can use classical dimensions for the query and the AP-attribute at the same time. Let suppose we have an hierarchy over *home town* attribute and we grouped the values as follows:

{Granada county, Malaga county, Jaen county, Rest of Spain, Abroad}

If we choose again the *count* aggregation the result for weak matching and strong matching are shown in Tables 5 and 6 respectively. A example using a different aggregation function is shown in Table 7, using the *average* to aggregate the *waiting time* .

6 Conclusions

In this paper we have presented a multidimensional model that supports the use of textual information in the dimensions by means of a semantical structures called AP-structures. To build these structure, a process is carried out so these AP-structure represent the meaning behind the text instead of a simple set of terms. The using of the AP-structure inside the multidimensional model enrich the OLAP analisys so the user may introduce the sematic of textual attribute in the queries over the datacube.

To complete the model we need to provide the dimension associated to the AP-attribute with the normal operation over a hierarchy allow the user to choose different granularities in the detail levels. All these extension to the multidimensional will be integrated inside an OLAP system to build a prototype a test the behaviour of the proposal with real databases.

No.	Waiting time	Town	spanning set of AP-attribute
1	10	Granada	(pain,leg)
2	5	Gojar	(pain head), (vomit)
3	10	Motril	(pain,head), (vomit)
4	15	Granada	(fracture) (vomit)
5	15	Armilla	(pain,intense, head)
6	5	Camaguey	(pain, intense, leg)
7	5	Málaga	(pain, leg)
8	5	Sevilla	(pain,head)
9	10	Sevilla	(pain), (stomach)
10	5	Gojar	(fracture)
11	10	Granada	(fracture leg)
12	5	Santafé	(fracture) (head)
13	5	Madrid	(vomit, stomach)
14	5	Madrid	(vomit, stomach)
15	12	Jaen	(pain, intense, leg)
16	15	Granada	(pain, intense, leg)
17	5	Motril	(pain, intense, head)
18	10	Motril	(pain, intense)
19	5	London	(fracture, leg)
20	15	Madrid	(pain, intense), (vomit, stomach)

Table 2. Database after the process

(pain intense)	(vomit)	Other	Total
13	6	4	23

Table 3. One dimension datacube using weak matching

(pain intense)	(vomit)	Other	Total
7	6	8	21

Table 4. One dimension datacube using strong matching

	(pain intense)	(vomit)	Other	Total
Granada c.	7	3	3	13
Malaga c.			1	1
Jaen c.	1			1
Rest of Spain	3	3	0	6
Abroad	1		1	2
Total	13	6	4	23

Table 5. Two dimensions datacube using weak matching

	(pain intense)	(vomit)	Other	Total
Granada c.	4	3	4	11
Malaga c.			1	1
Jaen c.	1			1
Rest of Spain	1	3	2	6
Abroad	1		1	2
Total	7	6	8	21

Table 6. Two dimensions datacube using strong matching

	(pain intense)	(vomit)	Other	Total
Granada c.	11.5	10	7.5	9.3
Malaga c.			5	5
Jaen c.	12			12
Rest of Spain	15	6.5	7.5	9.6
Abroad	5		5	5
Total	10.8	8.3	6.6	

Table 7. Two dimensional datacube using strong matching and average time aggregation

Bibliography

[1] R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases, 1995.

[2] A. Datta and H. Thomas. The cube data model: A conceptual model and algebra for on-line analytical processing in data warehouses. *Decision Support Systems*, 27:289–301, 1999.

[3] S. Keith, O. Kaser, and D. Lemire. Analyzing large collections of electronic text using olap. Technical report, In APICS 2005, 2005.

[4] R. Kimball. *The Data Warehouse Toolkit*. Wiley, 1996.

[5] N. Marín, M. J. Martín-Bautista, M. Prados, and M.A. Vila. Enhancing short text retrieval in databases. In *Proceedings of FQAS*, Milan, Italy, June 2006.

[6] M. J. Martín-Bautista, S. Martínez-Folgoso, and M. A. Vila. A new semantic representation for short texts. In *To appear in Proceedings of the 10th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2008), LNCS Springer-Verlag*, Turin, Italy, September 2008.

[7] M. J. Martín-Bautista, M. Prados, M. A. Vila, and S. Martínez-Folgoso. A knowledge representation for short texts based on frequent itemsets. In *Proceedings of IPMU*, Paris, France, 2006.

[8] C. Molina, L. Rodríguez-Ariza, D. Sánchez, and M. A. Vila. A new fuzzy multidimensional model. *IEEE T. Fuzzy Systems*, 14(6):897–912, 2006.

[9] T. Niemi, M. Niinimki, J. Nummenmaa, and P. Thanisch. Applying grid technologies to xml based olap cube construction. In *In Proc. DMDW03*, pages 2003–004, 2003.

[10] F. S. C. Tseng and A. Y. H. Chou. The concept of document warehousing for multi-dimensional modeling of textual-based business intelligence. *Decision Support Systems*, 42(2):727–744, 2006.