A new multidimensional model with text dimensions: definition and implementation

Maria J. Martin-Bautista*[1], Carlos Molina [2], Elizabet Tejeda[3], Maria-Amparo Vila [1]

[1] *Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain*
E-mail: {mbautis,vila}@decsai.ugr.es

[2] *Department of Languages and Computer Systems, University of Jaen, Jaen, Spain*
E-mail: carlosmo@ujaen.es

[3] *Department of Computers, University of Camagüey, Camagüey, Cuba*
E-mail: elizabeth.tejeda@reduc.edu.cu

Received 16 February 2011
Accepted 18 July 2012

Abstract

We present a new multidimensional model with textual dimensions based on a knowledge structure extracted from the texts, where any textual attribute in a database can be processed, and not only XML texts. This dimension allows to treat the textual data in the same way as the non-textual one in an automatic way, without user's intervention, so all the classical operations in the multidimensional model can been defined for this textual dimension. While most of the models dealing with texts that can be found in the literature are not implemented, in this proposal, the multidimensional model and the OLAP system have been implemented in a software tool, so it can be tested on real data. A case study with medical data is included in this work.

Keywords: Data warehousing, textual data, knowledge representation, textual dimensions.

1. Introduction

Data Warehousing (DW) and OLAP (On-line Analytical Processing) systems allow the analysis of multidimensional data, and have become strategic elements in companies nowadays for decision making. These systems usually work with discrete attributes with a well defined domain in databases. However, the management of textual information inside these systems is complex due to the lack of structure and the heterogeneity of textual data. In particular, as far as we know, there exist no implementations of Data Warehousing and OLAP able to analyze textual attributes in databases from a semantical point of view.

In this work, we show a multidimensional model with semantic treatment of text data and an implementation of Data Warehousing and OLAP systems using this model. The semantic aspect of this model comes from a knowledge structure of the data extracted before the DW process. The textual data can be obtained from external files or from textual at-

*Corresponding author: Maria J. Martin-Bautista. email: mbautis@decsai.ugr.es. C/ Daniel Saucedo Aranda s/n, 18071, Granada, Spain. Tlf: +34-958240805; Fax: +34-958243317 tributes in a database. The main objective of this proposal is to handle and query the textual data in the same way that we manage the non-textual one. This implies the datacube definitions and the multidimensional operations integrating both textual and non-textual data in an automatic and homogeneous way. For this purpose, we propose a new multidimensional model with textual dimensions. These textual dimensions are obtained through Text Mining techniques, from an intermediate representation form called AP-Structure [1]. This knowledge structure can be obtained automatically and keep the semantics of the texts [2].

The paper is organized as follows: after a brief description of the classical multidimensional model and a motivating example in the next Section, the literature related to our proposal, specially those approaches of Data Warehousing with texts, can be found in the Section 2. Preliminary concepts about the multidimensional model and the knowledge structure are given in Section 3. In Section 4, the formal multidimensional model with textual dimensions is described. The query process to the knowledge structure and its operations are presented in Section 5, while the Dice operation for textual dimensions is presented in section 6. A real application of the model to a medical data warehouse is shown in Section 7. Finally, the main conclusions are given in Section 8.

2. 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 [3],[4], since we do not consider that there is a standard one [5]. This model is the base of most of the proposals reviewed in Section 3, 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 $D_i = \{x_1 \ldots x_{m_i}\}$ of discrete values so, each tuple t of the database takes a unique and well determined value x_i in the attribute d_i. Let us note $d_i[t] = x_i$.
  - A grouping hierarchy that allows us to consider different values for the analysis. Such a hierarchy $\mathscr{H}_i = \{\mathscr{C}_{i1} \ldots \mathscr{C}_{il}\}$ is formed by partitions of D_i in a way that:

$$\forall k \in \{1, 2 \ldots l\}\ \mathscr{C}_{ik} \subseteq \mathscr{P}(D_i) \qquad (1)$$

$$\mathscr{C}_{ik} = \{X_{ik}^1, .., X_{ik}^h\}$$

  - being

$$\forall j, r\ X_{ik}^j \bigcap X_{ik}^r = \emptyset \text{ and } \bigcup_{j=1}^{h} X_{ik}^j = D_i \qquad (2)$$

  The hierarchy $\mathscr{H}_i$ is an inclusion reticulum which minimal element is the partition of D_i, which contains each of its elements considered as an isolated set, and the maximal is the complete 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 $\mathscr{C}_{ik}$ in the dimension d_i, then $Y_i \in \mathscr{C}_{ik}$. 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 exists also an aggregation criterion of V, *AGG*, which is applied when 'set' values are considered in any of the dimensions. That is,

$$V = f(x_1, ..Y_k, ..x_n) = \qquad (3)$$

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

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

Example 1 ***Motivating example***
As an example, let us consider a simplification of patient data in an emergency service in a hospital. Table 1 shows some records stored in the database. Attributes *Patient number (no)*, *Age*, *Admission Date*, *Town* are classical attributes. *Diagnosis* is a free textual attribute that stores the information given by the medical doctor for the patient.

Table 1 and Table 2 show how the multidimensional model is built from a database sample. Let us consider the dimensions admission-date and town, being the hierarchies for these the following ones:

- For admission-date:
  - Week-day={Mon, Tue, Wed, Thu, Fri, Sat, Sun} that is also grouped into Day-type={working-day={Mon, Tue, Wed, Thu, Fri}, holiday{Sat, Sun}}
  - Month={01,02,...,12} that is also grouped into Season={Spring{03,04,05}, Summer{06,07,08}, Autumn{09,10,11}, Winter{12,01,02}}
  - Year={2008, 2009}
- For town:
  - Region
  - Situation={South-east, South-west, Center, East, North-east, North-west}

From the concept of data cube, the normal operations are defined. They correspond to the different possibilities of analysis on the dimensions. The most common operations are:

- **Roll-up** resumes the cube data 'ascending' in the hierarchies of a dimension from a more specific partition to a more generic one. Considering, for instance the tuple counting as measure, grouping the months by seasons in the time dimension, the obtained cube would be as in Table 2.
- **Drill-down** is the opposite operation, 'descending' from a more generic partition to a more specific one.
- There are other operations like **Slice** and **Dice** that restrict the information to a part of the database.

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 [6].

In the cube definition and their operations, we consider that the dimensions are defined on domains of well established discrete values, where the hierarchies are built on. If we consider some other kind of domains, this concept needs to be extended. The problem with the management of attributes with textual information is specially hard to deal with, since there is a lack of structure, the semantics has to be considered and the automatic process of this kind of information is not easy to perform without user's intervention. In the following, we resume the most relevant works found in the literature to solve this problem, and we give the devotion of our proposal.

3. Related Work

In the literature, we can found several proposals of Data Warehousing in relation with textual data. Different techniques are used in these proposals 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. For instance, the works of [7], [8],[9], [10] and [11] are centered in the use of the advantages of the XML to analyze the explicit information of textual documents. They consider only external textual documents as sources of data, but they ignore the abundant textual information of attributes in databases. In some studies as [12] and [13], they expose the possibility to pre-process the textual documents with the help of techniques from Data Mining and Information Retrieval, to extract knowledge of these, and to analyze the same in multidimensional specific models. The work of Ravat proposes new measures based on texts, but they do not process the texts contained in the dimensions.

n-patient	**age**	**admission-date**	**town**	**diagnosis**
1	25	12/02/2009 Fri	Granada	pain in left leg
2	20	15/03/2009 Thu	Córdoba	head pain and frequent vomits
3	10	10/06/2009 Sun	Albacete	head pain and vomits
4	50	08/02/2009 Mon	Álava	fracture and vomits
5	3	10/10/2009 Wed	Madrid	intense pain in the head
6	12	05/03/2009 Mon	Badajoz	intense pain in leg
7	65	07/12/2009 Fri	Madrid	pain in right leg
8	45	06/11/2009 Tue	Murcia	mild pain in the head
9	70	05/01/2009 Fri	Jaén	stomach pain and sickness
10	18	05/05/2009 Sat	Sevilla	foot fracture
11	8	01/06/2009 Fri	León	fracture in left leg
12	70	25/10/2009 Thu	Salamanca	fracture in the head
13	30	30/03/2009 Fri	Toledo	vomits and stomach acidity
14	20	31/12/2009 Mon	Madrid	vomits and stomach flatulence
15	26	04/11/2009 Sun	Valencia	intense pain of leg and hip
16	44	11/07/2009 Wed	Castellón	intense pain in the leg and in the forearm
17	35	13/08/2009 Mon	Sevilla	intense pain head and loss of vision
18	75	14/06/2008 Wed	Málaga	intense pain in the arms
19	8	08/08/2008 Tue	Córdoba	leg fracture
20	12	12/12/2008 Tue	Alicante	intense pain in the stomach and vomits

Table 1: Part of a table from a medical database

	South-east	South-west	Center	East	Nor-east	Nor-west	Total
Spring		2	2				4
Summer	1	2	1	1		1	6
Autumn			2	2			4
Winter	2		2	1	1		6
Total	3	4	7	4	1	1	20

Table 2: Counting measure with the hierarchies of date in seasons and of town in situation

In [14], [15] and [16] they start from multidimensional textual databases, and applying mining of data and/or texts, they also process the textual information and propose new models of OLAP cubes. In these works, they intend to process textual dimensions to obtain new numeric measures. It can be said that these measures limit the analysis of these dimensions in a combined way with other classic measures such as the sum or the average, very widely used in decision making.

Other proposals combine independent tools, of mining and OLAP to be able to analyze the results of mining, like in [17], [18], [19].

It can be appreciated that in these last works, they mostly refer to the extraction of keywords [20], [21], and not to the extraction of another type of knowledge from the texts. The proposal of Perez et al. for the data warehousing of contexts of documents, Ravat's works for the textual measure that they propose and Inokuchi's works for the use of external ontologies to create hierarchies in the dimensions are specially remarkable. In [22] a method to carry out OLAP over textual data is proposed. For this purpose, they present a model for the representation of the data, with their corresponding algebraic operations for the traditional OLAP operations. This model allows to integrate semantic information (in this case by means of the use of ontologies) in OLAP systems. This allows to analyze a great group of textual documents with their underlying semantic information. It is necessary to point out that the ontologies mentioned before are external to the processed data and obtained previously. It happens in a similar way in [23] and [24]. This can lead some of the users to not have any answering data for their queries.

The advantages of our proposal in comparison to the works above are:

- Any textual attribute in a database can be processed. Actually, any entry data can be also processed, once is transformed into an attribute in a database. This approach does not limit the source data to XML files, as most of the proposals in the literature do, taking into account that mostly the use of XML data implies the intervention of the user to generate and structure them.

- The processing of the textual attribute in the database is not a syntactic transformation, since it takes the semantics of the text and the resulting knowledge structure is directly understandable by the user. This knowledge structure is based on the concepts of AP-Set and AP-Structure, that have been revealed as a possible valid solution for the automatic representation of short textual data [1], [2].
- This transformation from the original textual attribute to the knowledge structure attribute can be obtained automatically without the user's intervention. Moreover, the knowledge used for this transformation comes from the database, and not from external data as in most of the proposals in the literature.
- In the cube processing, some of the works propose a textual measure, but they do not value as we do, the analysis of the texts contained into the dimensions, since due to the semantic treatment of the textual data, a semantic dimension in the data cube is generated.
- Data Warehousing and OLAP processes are then implemented and performed in a real tool, while other proposals only define the model without any implementation. Moreover, our model and tool can be used in whatever the data field.

Moreover, the tool has been tested with real data from a medical database of the Clinical Hospital San Cecilio in Granada, Spain. In the following we present some preliminary concepts. With the purpose of a best reading of the work, we include some illustrative examples in the medical domain.

4. Preliminary concepts

This section contains some background needed for a best understanding of the proposal. The main definitions, operations and properties of the knowledge structure of representation for textual data are given. This representation called AP-Structure is detailed in previous works [1], [2], and it is as a possible valid solution for the automatic representation of short textual data.

4.1. AP-Structure: Knowledge structure of representation for textual data

Firstly, we will establish the definition and properties of the sets of subsets which have the "a priori" property (AP-Sets). Next, we will give the formal definition and properties of the structure underlying in the texts which is that of a set of AP-Sets. Previous definitions and a more detailed study of these structures can be found in [1, 2, 25].

4.2. AP-Set definition and properties

Definition 1 ***AP-Set***

Let be $X = \{x_1 ... x_n\}$ a finite set of items and $\mathscr{R} \subseteq \mathscr{P}(X)$ a set of frequent itemsets, being $\mathscr{P}(X)$ the set parts of X. We will say $\mathscr{R}$ is an AP-Set if and only if:

1. $\forall Z \in \mathscr{R} \Rightarrow \mathscr{P}(Z) \subseteq \mathscr{R}$
2. $\exists Y \in \mathscr{R}$ such that :
   (a) $card(Y) = max_{Z \in \mathscr{R}}(card(Z))$ and does not exist $Y' \in \mathscr{R}, Y' <> Y$, so that $card(Y') = card(Y)$
   (b) $\forall Z \in \mathscr{R}; Z \subseteq Y$

The set Y of maximal cardinal characterizes the AP-Set and it will be called *spanning set of* $\mathscr{R}$. We will denote $\mathscr{R} = g(Y)$, 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, the AP-Set of level equal to 1 contains single elements of X; we will consider the empty set $\emptyset$ as the AP-Set of zero level.

Example 2 *Let be* $X = \{pain, leg, head, vomit, fracture, intense, stomach\}$ *and* $\mathscr{R} = \{\{leg\}, \{intense\}, \{pain\}, \{leg, intense\}, \{pain, intense, leg\}\}$, *the spanning set is* $Y = \{pain, intense, leg\}$

It should be remarked that the definition 1 implies that any AP-Set $g(Y)$ is in fact the reticulum of $\mathscr{P}(Y)$

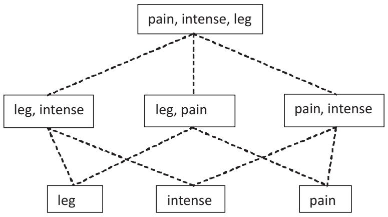

Figure 1: AP-Set reticulum

Figure 1 shows the reticulum corresponding to the example 2

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

$$\mathscr{R} \subseteq \mathscr{S} \Leftrightarrow R \subseteq S \tag{4}$$

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

$$\mathscr{S} = g(R \bigcap Y) \tag{5}$$

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

$$\mathscr{V} = g(R \bigcup Y) \tag{6}$$

4.3. 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 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 is formally defined as follows.

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

$$\forall A, B \in S; A \not\subseteq B, B \not\subseteq A \tag{7}$$

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

Next, one of the most important operations concerning AP-Structures is introduced, that of the restriction of an AP-Structure by a set to give a new induced AP-Substructure. This operation is particularly relevant because it will allow us to carry out the process described in 4.5, by obtaining the restriction of the global database structure to the term set associated with each tuple of the database.

Definition 6 Induced AP-Substructure

Let be an AP-Structure $\mathscr{T} = g(A_1, A_2, ..., A_n)$ *with referential set of items X and* $Y \subseteq X$*. We will define* AP-Substructure of $\mathscr{T}$ induced by Y*:*

$$\mathscr{T}' = \mathscr{T} \bigwedge Y = g(B_1, B_2, ..., B_m)$$

where

$$\begin{gathered}\forall B_i \in \{B_1, ..., B_m\} \Rightarrow \exists A_j \in \{A_1, A_2, ..., A_n\} \\ \textit{so that } B_i = A_j \bigcap Y \\ \forall A_j \in \{A_1, ..., A_n\} \Rightarrow \exists B_i \in \{B_1, B_2, ..., B_m\} \\ \textit{so that } A_j \bigcap Y \subseteq B_i\end{gathered}$$

Following this definition, it is clear that $\mathscr{T}'$ is the AP-Structure generated by the intersections of Y with the spanning sets of $\mathscr{T}$. However, the resulting intersections among the spanning sets of $\mathscr{T}$ and Y that are already included in other set B_i will be eliminated. In this way, it is possible to guarantee that the obtained AP-Structure is only formed by maximal spanning sets, following the defined AP-Structure concept. The following example clarifies this point:

Example 3 Let be $X = \{pain, leg, head, vomits, fracture, intense, stomach\}$,

$\mathscr{T} = g(\{vomits, stomach\}, \{fracture, leg\}, \{pain, intense, leg\}, \{pain, intense, head\})$,

$Y = \{fracture, leg, pain\}$, then we have

$\mathscr{T} \bigwedge Y = g(\{fracture, leg\}, \{leg, pain\})$.

4.4. *Matching sets with AP-Structures*

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

Two approaches are proposed: *weak* and *strong matching*. A detailed definition can be found in [25, 2]. The idea behind the matching is to compare the spanning sets for the AP-Structure and the set of terms given by the user. The *strong matching* considers that the set of terms by the user and the AP-Structure match if all the terms are included in a spanning set. The *weak matching* relaxes the condition and returns *true* if at least one of the terms is included in a spanning set.

Definition 7 ***Strong matching***
Let be an AP-Structure $\mathscr{T} = g(A_1, A_2, ..., A_n)$ with referential X and $Y \subseteq X$; we define the *strong matching between Y and $\mathscr{T}$* as a logical operation:

$$Y \bigodot \mathscr{T} = \begin{cases} \text{true if} & \exists A_i \in \{A_1, A_2, ..., A_n\} / Y \subseteq A_i \\ \text{false} & \text{otherwise} \end{cases} \tag{8}$$

Definition 8 ***Weak matching*** *Let be an AP-Structure $\mathscr{T} = g(A_1, A_2, ..., A_n)$ with referential X and $Y \subseteq X$; we define the* weak matching between Y and $\mathscr{T}$ *as a logical operation:*

$$Y \bigoplus \mathscr{T} = \begin{cases} \textit{true if} & \exists A_i \in \{A_1, A_2, ..., A_n\} / \\ Y \bigcap A_i \neq \emptyset & \\ \textit{false} & \textit{otherwise} \end{cases} \tag{9}$$

These matching criteria 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 a larger index than other with less terms. Additionally, if some term set matches with more than one spanning set, it will have a larger index than that of the other one which only matches one set. Obviously, two matching indexes can be established, but both will have similar definitions.

Definition 9 ***Strong(weak) matching index***
Let be an AP-Structure $\mathscr{T} = g(A_1, A_2, ..., A_n)$ with referential X and $Y \subseteq X$, we define the *strong(weak) matching index between Y and $\mathscr{T}$* as follows:

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

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

$$\textbf{Strong index} = S(Y|\mathscr{T}) = \sum_{i \in T} m_i(Y)/n \tag{10}$$

$$\textbf{Weak index} = W(Y|\mathscr{T}) = \sum_{i \in V} m_i(Y)/n \tag{11}$$

Obviously:

$$\forall Y \text{ and } \mathscr{T}\ ,\ S(Y|\mathscr{T}) \in [0,1]\ ,\ W(Y|\mathscr{T}) \in [0,1] \\ \text{and } W(Y|\mathscr{T}) \geqslant S(Y|\mathscr{T}) \tag{12}$$

4.5. *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*. Further details of this process can be found in [1], [2].

1. The frequent terms associated to the textual attribute are obtained. This process includes a cleaning process, an empty word deleting process, synonymous management process using dictionaries, etc. Then, we get a set of basic terms T to work with. In this point, the value of the textual attribute on each tuple t is a subset of basic terms T_t. This consideration allows us to work with the tuples as in a transactional database regarding the textual attribute.

2. Maximal frequent itemsets are calculated. Let be $\{A_1,..,A_n\}$ the itemsets, then the AP-Structure $S = g(A_1,..,A_n)$ includes all the frequent itemsets, so we can consider the AP-Structure to cover the semantics of the textual attribute.

3. Once we have the global AP-Structure, we obtain the AP-Structure associated to a tuple t: if T_t is the set of terms associated to t, the value of AP-attribute for the tuple is:

$$S_t = g(A_1,..,A_n) \bigwedge T_t \tag{13}$$

This process obtains the domain for any AP-attribute. We must remark that this process is performed off-line, so the execution time is not a decisive factor, although the complexity basically lays on the algorithm to obtain the frequent itemsets, following an Apriori like algorithm [26]. Moreover, a study about the scalability of the mining process can be found in [2], where is shown that the underlaying AP-Structure is quite robust against updates of the original database.

Definition 10 ***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

$$D_A = \{R = g(B_1,..B_m), /, \forall i \in \{1,..,m\}, \tag{14}$$

$$\exists j \in \{1,..,n\} \text{such that} B_i \subseteq A_j\}$$

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 the attribute A according to the previous constraint.

After applying the proposed process, we transform the textual attribute into an AP-attribute. Figure 2 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 store the spanning sets associated to each record, as shown in Table 3.

5. Formal Model

In this section, we define the main concepts and operations to describe the new multidimensional model with text dimensions.

5.1. 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 first need some considerations.

- Although the internal representation of an AP-attribute is a structure, the input and output for the user is carried out by means of terms sets ("sentences"), that 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 10, we are working with a structure domain and closed when we consider the union. So, a set of elements of the domain is included 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 11 ***AP-Structure partition associated to a query***

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

$$\mathscr{P} = \{S_1,..,S_q,S_{q+1}\} \tag{15}$$

where

$$S_i = \begin{cases} S \bigwedge T_i & \text{if } i \in \{1,..,q\} \\ S \bigwedge (X - \bigcup_{i=1}^{q} T_i) & \text{otherwise} \end{cases} \tag{16}$$

Now we can introduce a multidimensional model as it was defined in Section 3, which uses an AP-Dimension:

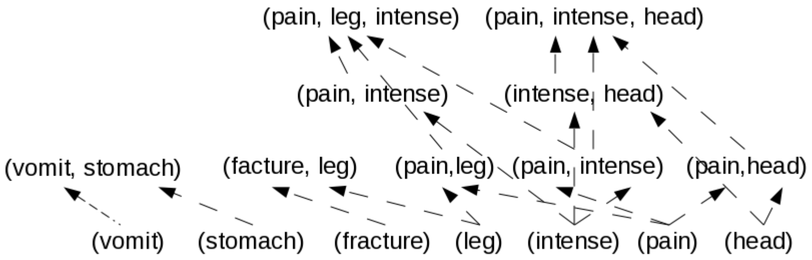

Figure 2: Global AP-Structure

n-patient	age	admission-date	town	diagnosis
1	25	12/02/2009 Fri	Granada	(pain, leg)
2	20	15/03/2009 Thu	Córdoba	(pain head), (vomit)
3	10	10/06/2009 Sun	Albacete	(pain,head), (vomit)
4	50	08/02/2009 Mon	Álava	(fracture) (vomit)
5	3	10/10/2009 Wed	Madrid	(pain,intense, head)
6	12	05/03/2009 Mon	Badajoz	(pain, intense, leg)
7	65	07/12/2009 Fri	Madrid	(pain, leg)
8	45	06/11/2009 Tue	Murcia	(pain,head)
9	70	05/01/2009 Fri	Jaén	(pain) (stomach)
10	18	05/05/2009 Sat	Sevilla	(fracture)
11	8	01/06/2009 Fri	León	(fracture leg)
12	70	25/10/2009 Thu	Salamanca	(fracture) (head)
13	30	30/03/2009 Fri	Toledo	(vomit, stomach)
14	20	31/12/2009 Mon	Madrid	(vomit, stomach)
15	26	04/11/2009 Sun	Valencia	(pain, intense, leg)
16	44	11/07/2009 Wed	Castellón	(pain, intense, leg)
17	35	13/08/2009 Mon	Sevilla	(pain, intense, head)
18	75	14/06/2008 Wed	Málaga	(pain, intense)
19	8	08/08/2008 Tue	Córdoba	(fracture, leg)
20	12	12/12/2008 Tue	Alicante	(pain, intense), (vomit, stomach)

Table 3: Table of the database after the process

- $\forall i \in \{1,..,q\}$ $V = f(C_1,...,S_i,...C_n)$ is a function aggregation (count, or other numeric aggregation) associated with the tuples that satisfy T_i in any way.
- $V = f(C_1,...,S_{q+1},...C_n)$ is a function aggregation associated to the tuples not matching any sentences in T_i, or part of them, which means, the sentences that are not related with the sentences given by the user.

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

Example 4 *Considering Table 3, we can define the partition associated to query C:* $C = \{(intense, pain), (vomits)\}$

Then, considering the counting as aggregation measure and the weak matching as matching criteria, we obtain the cube in Table 4. Analogously, we obtain the cube in Table 5 applying the strong matching as matching criteria.

6. Operations over an AP-Structure associated to a query

We have defined the form that have the partitions of the AP-Structure and how they are generated. The following step is to define the operations on the same ones. For it, the first thing will be to establish a hierarchy of partitions of a AP-Structure. Obviously, since the partition of a AP-Structure is associated to a query, it is clear that the hierarchy of partitions will

(intense, pain)	(vomits)	Other	Total
13	6	4	23

Table 4: Weak matching example for a partition associated to a query

(intense, pain)	(vomits)	Other	Total
7	6	8	21

Table 5: Strong matching example for a partition associated to a query

be associated to a hierarchy of queries.

6.1. *Definition of hierarchy of queries*

Definition 12 ***Hierarchy of an AP-Structure partition associated to a query***

Let be $C^1 = \{T_1^1, .., T_q^1\}$ and $C^2 = \{T_1^2, .., T_n^2\}$ two sets of ”sentences” over X; that means, all the possible queries over a dimension. Then the query C^1 is more detailed than C^2 and we note it sd $C^1 << C^2$ if and only if:

$$\begin{gathered} C^1 << C^2 \Leftrightarrow \forall T_i^{1 \in C^1}\ \exists T_j^{2 \in C^2} / T_j^2 \subseteq T_i^1 \quad \textit{and} \\ \forall T_j^2\ \exists T_i^1 / T_j^2 \subseteq T_i^1 \quad \textit{and} \\ \exists T_i^1 / T_i^1 \subseteq T_j^2 \end{gathered}$$

Intuitively, a query is more detailed than another when the first one contains ”more detailed sentences” than the second. Let show an example:

Example 5 *Let be $C^1 = \{(pain, head), (fracture)\}$ and $C^2 = \{(pain), (fracture)\}$*

We get $C^1 << C^2$, since it satisfies:

$$\begin{gathered} \forall T_i^{1 \in C^1}\ \exists T_j^{2 \in C^2} / T_j^2 \subseteq T_i^1 \quad \textit{and} \\ \forall T_j^2\ \exists T_i^1 / T_j^2 \subseteq T_i^1 \quad \textit{and} \\ \exists T_i^1 / T_i^1 \subseteq T_j^2 \end{gathered} \tag{17}$$

So, we observe that each set belonging to C^2 appears in C^1.

The following property allows to extend the idea of a ”more detailed” partition to the underlying AP-Structures.

Property. Let be $C^1 = \{T_1^1, .., T_q^1\}$ and $C^2 = \{T_1^2, .., T_h^2\}$ two queries over X; such that $C^1 << C^2$. Let be

$$\begin{gathered} P^1 = \{S_1^1, ..S_q^1, S_{q+1}^1\} \\ P^2 = \{S_1^2, ..S_h^2, S_{h+1}^2\} \end{gathered} \tag{18}$$

The sub-AP-Structures associated to the queries satisfy the following properties:

1. $\forall i \in \{1, .., q\}\ \exists j \in \{1, 2..h\}$ such that $S_j^2 \subseteq S_i^1$
2. $\forall j \in \{1, .., h\}\ \exists i \in \{1, 2..q\}$ such that $S_j^2 \subseteq S_i^1$
3. $S_{h+1}^2 \supseteq S_{q+1}^1$

Let us prove each property.

1. Property 6.1.1

   Let be $S_j^2 \in P^2$ according to the definition of partition associated to a query
   $\exists T_j^2 \subseteq C^2$ such that $S_j^2 = S \bigwedge T_j^2$
   As they satisfy $C^1 << C^2$ then we get $\exists T_i^1$ / $T_j^2 \subseteq T_i^1$; for T_i^1 it has a sub-AP-Structure associated S_i^1 defined as:
   $S_i^1 = S \bigwedge T_i^1$
   been $T_j^2 \subseteq T_i^1 \Rightarrow T_i^1 \cap T_j^2 = T_j^2$
   we have:
   $S_j^2 = S \bigwedge T_j^2 = S \bigwedge (T_i^1 \cap T_j^2)$ and considering the AP-Structure properties ([27]) we have:
   $S \bigwedge (T_i^1 \cap T_j^2) = (S \bigwedge T_i^1) \bigwedge T_j^2$ and by definition
   $(S \bigwedge T_j^2) \supseteq (S \bigwedge T_i^1) \bigwedge T_j^2$ that means
   $S_i^1 \supseteq S_j^2$ what prove this property.

2. Property 6.1.2 Let be $S_i^1 \in P^1$ according to the partition associated to a query, $\exists T_i^1 \in C^1$ such that:
   $S_i^1 = S \bigwedge T_i^1$
   considering $C^1 << C^2$ from definition 12 we get:
   $\exists T_j^2 \in C^2$ / $T_i^1 \supseteq T_j^2$
   and for T_j^2 we have the following associated sub-AP-Structure: $S_j^2 = S \bigwedge T_j^2$

using the same reasoning as in the previous case, we get:
$S_j^2 \subseteq S_i^1$

3. Property 6.1.3 According to definition
$S_{q+1}^1 = S \bigwedge (X - \bigcup_{i=1}^q T_i^1) = S \bigwedge P_1$
$S_{h+1}^2 = S \bigwedge (X - \bigcup_{j=1}^h T_j^2) = S \bigwedge P_2$
considering the definition of finest partition, we have:
$P^2 \supseteq P_1$
Let be:

$W_1 = \bigcup_{i=1}^q T_i^1$; $W_2 = \bigcup_{j=1}^h T_j^2$

it satisfies $W_2 \subseteq W_1$ because in other case we get:

$\exists a \in X$ such that $a \in W_2$ y $a \notin W_1$. Let be T_j*^2 / $a \in T_j*^2$, it considering $a \notin W_1$ there is no T_i^1 that contains T_j*^2 , what is against the definition of more detailed partition.
Obviously if $W_2 \subseteq W_1 \Rightarrow X - W_2 \supseteq X - W_1$ or equivalently $P_2 \supseteq P_1$.
Considering this, we get:
$S_{h+1}^2 \supseteq S_{q+1}^1$

Property 6.1 has important consequences because the translation of the property is the following: the whole query hierarchy defined by a "more detailed than" $<<$ induces a hierarchy with a sequence of sub-AP-Structure hierarchy that verifies the inclusion relationship. If we add an AP-Structure to the sequences that transform it in a partition, the inclusion relationship is the inverse.

The query hierarchy definition allows us to define the Roll-Up and Drill-Down operations over an AP-Dimension.

Let be C^1 a query and P^1 an associated partition:

- We apply Roll-Up over C^1 if we consider and query C^2 such that $C^1 << C^2$ and the associated partition P^2. That means, we consider a query with "less precise sentences".
- We apply Drill-Down if we consider a query C^3 such that $C^3 << C^1$ and its partion P^3. That means, we have a query with more detailed sentences than C^1.

In Figure 3, an example of hierarchy of queries with three levels and the Roll-Up and Drill-Down operations is shown.

6.2. Matching

Let be C^1 and C^2 two queries over X such that $C^1 << C^2$
where $C^1 = \{T_1^1, .., T_q^1\}$ and $C^2 = \{T_1^2, .., T_h^2\}$ then

$$P^1 = \{S_1^1, ..S_q^1, S_{q+1}^1\}$$
$$P^2 = \{S_1^2, ..S_h^2, S_{h+1}^2\}$$

are the associated partition to both queries. If we consider a matching (strong or weak) and an count measure m associated to the queries, we get:

$$m_1^1, ..m_q^1, m_{q+1}^1 \tag{19}$$

$$m_1^2, ..m_h^2, m_{h+1}^2 \tag{20}$$

as associated values to the matching, where:

$$\forall i \in \{1, .., q\} \exists j \in \{1, 2..h\} / m_i^1 \leqslant m_j^2 \tag{21}$$

$$\forall j \in \{1, .., h\} \exists i \in \{1, 2..q\} / m_i^1 \leqslant m_j^2 \tag{22}$$

That means, we get a higher value in C^2 than in C^1 when using the same matching for both.

Let consider jS (strong matching and count j) for a level and jW (weak matching and count m) for another . Then if S is used for C^2 and W for C^1, then we get $i \in \{1, 2..q\}$ and $j \in \{1, 2..h\}$ such that:

$$T_i^1 \supseteq T_j^2$$

It is clear that: all tuples that satisfy the strong matching for T_j^2, then they satisfy the weak matching for T_i^1 then:

$$S_j^2 \leqslant W_i^1$$

That means, we have $\forall i \in \{1, .., q\}$ $\exists j$ such that

$$mS_i^1 \leqslant mS_j^2 \leqslant mW_i^1 \leqslant mW_j^2 \tag{23}$$

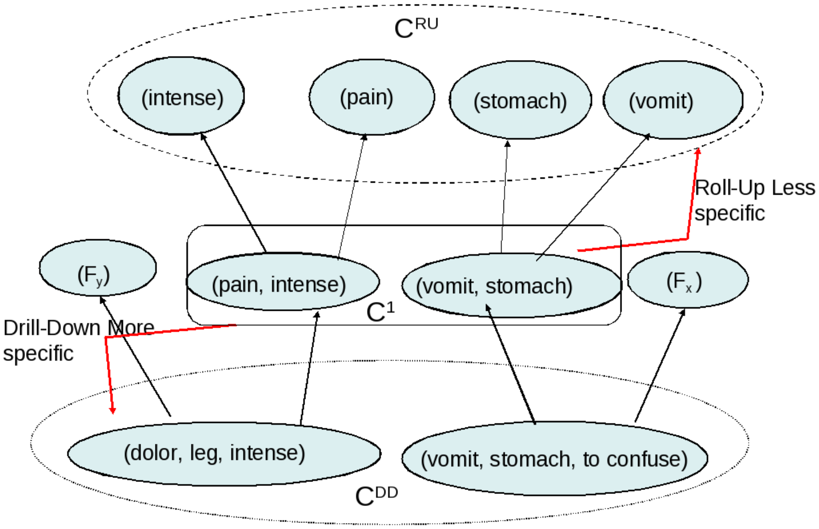

Figure 3: Roll-Up y Drill-Down examples over a hierarchy of queries

A particular case is when the queries have only one term (uniterm). In this case $mS_j^2 \equiv mW_i^1$; if T_j^2 is uniterm due to the fact that in this case

$$mS = mW$$

and then

$$mS_j^2 = mW_i^1 = mW_j^2. \qquad (24)$$

Next section presents the Dice operation for an AP-Dimension.

6.3. *Dice operation for an AP-Dimension*

Definition 13 ***Dice***

Let be an AP-Dimension with a global AP-Structure $S = \{(A_1, A_2, \ldots, A_n)\}$. Let be T a set of terms such that $T \subseteq \bigcup_{i=1}^{n} A_i$, we apply *Dice* over the AP-Dimension associated to T when the domain of the dimension over the global AP-Structure is restricted:

$$R = S \bigwedge T, \qquad (25)$$

we just consider $R = \{(A_1 \cap T, A_2 \cap T, \ldots, A_n \cap T)\}$ as global *AP-Structure*.

7. Operation examples

In order to reinforce the operations given above, some examples of the hierarchy, matching and Dice operation are given in the following.

7.1. *Hierarchy and matching examples*

Example 6 *Let be $C^1 = \{(pain, head), (fracture)\}$ y $C^2 = \{(pain), (fracture)\}$*

then P^1 and P^2 the associated partition to the queries, respectively

$$P^1 = \{(pain, intense, head), (fracture, leg),$$

$$(other)\}$$

$$P^2 = \{(pain, intense, leg), (pain, intense,$$

$$head), (fracture, leg), (other)\}$$

These partitions satisfy the three properties previously presented. One set of P^2 is included in P^1, and all the sets of terms in P^1 are included in P^2.

The resulting datacube when we query C^1 using strong matching is shown in Table 6.

If we want to know the number of diagnosis in a more or less detailed way, we can use the AP-hierarchy for C^1. Less and more detailed queries (C_{LD} and C_{MD}, respectively) are possible for C^1, where $C_{LD} = \{\{head\}, \{pain\}, \{fracture\}\}$ and $C_{MD} = \{pain, intense, head\}$

(pain, head)	(fracture)	Other	Total
5	5	10	20

Table 6: Strong matching example

(pain,intense,head)	Other	Total
2	18	20

Table 7: More detailed query using strong matching

The user can choose one or more sentences from the AP-hierarchy to build the query. In this example, the result if the user selects a more detailed query as $(pain, intense, head)$ is shown in Table 7.

In the other hand, we can use a less detailed query as $C^3 = \{(head)\}$. The result in this case is shown in Table 8.

Next, we present the relation between the operations and the two matching criteria. Table 9 shows the results of the query C^1 using weak matching. We can see that $jS^3_j \leqslant jW^1_i$ that was expected due to Equation 23.

7.2. *Dice example*

Let start from a cube with the following features:

- A textual dimension (AP-Dimension): *diagnosis*
- A classical dimension: *implants*
- Measure: *number of treatments*.
- Aggregation function: *sum*.

We can perform a Dice operation by the AP-dimension *diagnosis* to obtain the interventions related to the term *cataract*. The results of the operation are shown in Figure 4. Experts confirm that in most of the cataract surgery interventions, implants are used.

8. Implementation of the model: OLAP Wonder

OLAP Wonder [28], [29] implements the new multidimensional model exposed before, with support to free textual contents in the textual attributes of the database. OLAP Wonder is an OLAP server of free disposition, implemented with techniques and free software tools, that offers services of administration of OLAP cubes for any database in PostgreSQL.

The Wonder architecture has been designed based on the architecture proposed by Microsoft for the Client-Server application development in several layers [30]. Specifically, this architecture proposes a set of logical layers to divide the client application for a best performance and communication among all the software components. The interface layer allows to the users to see the cubes, the forms and the graphics generated by the system. The service layer contains all the services implemented for the system classes such as creating and refining the cubes. Finally, the data access layer gives access to the metadata, OLAP cubes and data sources. This layer communicates to three databases: the source database, from which the data are extracted to populate the cubes, the database where the models are stored and the database where the original and refined cubes are stored.

Some query procedures considered in OLAP Wonder are:

(head)	Other	Total
6	14	20

Table 8: Less detailed query C^3 using strong matching

(pain,head)	(fracture)	Other	Total
14	5	2	21

Table 9: Query C^1 using weak matching

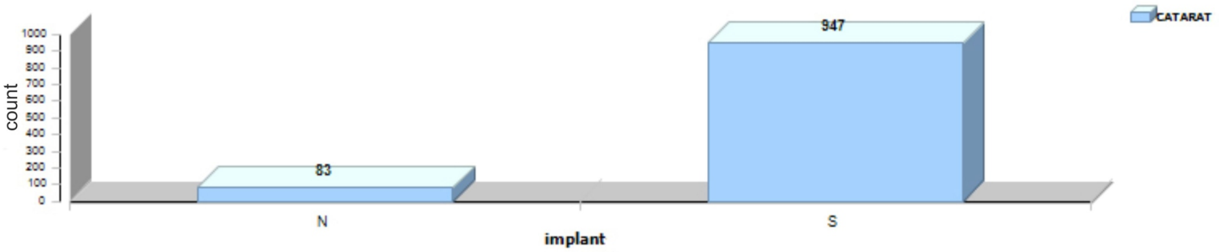

Figure 4: Number of implants in cataract interventions

- The hypercubes are constructed based on the multidimensional design of the model. Storing all the hypercube information, we can perform different operations following the user requirements.
- We can perform a Slice operation to eliminate the dimensions non related to the user query. A resulting subcube is then stored.
- The classical operations: Slice, Dice, Drill-Down, Roll-up as well as the use of the hierarchy associated to a query for the AP-dimensions can be performed over the original hypercube or over the generated subcube.

We must also remark the graphical capabilities de OLAP Wonder, so the results of queries can be represented in a bar, line or sector graphics, making easier to the user their understanding. The running examples in the next sections show some graphics emphasizing this OLAP Wonder feature.

8.1. Application to a medical data warehouse

We have carried out a complete implementation of the proposed model with data from several real cases. Here, we show the case study with real medical information. The data sets have been extracted from a hospital database of the “ San Cecilio” University Clinical Hospital of Granada, Spain. A preliminary study of the necessities of information has been made on some of the activities that are carried out in this hospital. As a result of that analysis a portion of the data has been extracted, specifically the information with respect to the Surgical Interventions(*TSurgical* table) and Emergencies (*TEmergencies* table). The content of these attributes is short (from one up to fifty terms), and they can include one or more sentences. Two main tools have been used for this work. The first one is the Text Mining Tool [27], [2], used for the preprocessing stage and the obtaining of the AP-Structure. With the help of this tool, applying a support of 0.01, we obtain those AP-attributes corresponding to the proposed fields of treatments (*IProposal* attribute) and the diagnoses (*Diagnosis* attribute) of tables *TSurgical* and *TEmergencies*, respectively. These AP-attributes are used to define the corresponding AP-Dimension. The second tool is OLAP Wonder defined above. These tools can operate over the same database. The Text Mining Tool keeps the AP-Structure updated so the OLAP Wonder tool can use always updated data.

The next examples show some queries to an AP-Dimension in the hospital environment. It is important to highlight that the analysis of the queries that were carried out was revised by a specialist in medicine that has collaborated with this investigation, so both the queries and the results make sense from a medical point of view.

8.2. Running examples

Example 7 *We want to know the number of interventions where the sentences* C={(SURGERY,ENDOSCOPIC),(BIOPSY)} *appear completely in the* treatment proposal *(AP-Dimension).*

The subcube for Example 7 has the following structure:

- A textual dimension (AP-Dimension): *treatment proposal*
- Measure: *number of interventions*.
- Aggregation function: *count*.

Figure 5 shows graphically the result of OLAP Wonder for the query. According to the graphic, *BIOPSY* is more frequent than the other sentence *(SURGERY, ENDOSCOPIC)*. This is logical since, from a medical point of view, the first treatment is less invasive and more frequently used. The second treatment is used in very specific cases.

The query can be refined using the hierarchy over the AP-Dimension so we can get a more or less detailed query than the original. Figure 6 collects the hierarchy for the AP-Dimension.

Using the hierarchy, we can look for a more detailed result. As an example, we can use a more detailed query such as *{(SURGERY,ENDOSCOPIC,NASAL), (BIOPSY,SKIN)}*. In this case, we have applied Drill-Down over the dimension to reduce the granularity. Figure 7 shows the result of OLAP Wonder in this case. The chart shows that there are less interventions in this case than the previous, which is normal from a medical point of view.

The next example combines an AP-Dimension with classical ones.

Example 8 *Get the number of interventions, and the anesthesia used, where* treatment proposal *satisfies the sentence* C={(LAPAROTOMY),(REMOVAL, CYST) (OSTEOSYNTHESIS)}.
The operations that we have to apply to get the results are the following:

- Hierarchy generation with the corresponding query using strong matching.
- *Slice* to consider only the dimensions *treatment proposal* and *anesthesia*.
- Aggregation of measure *number of interventions* using *count*.

The result of this query is a datacube with the following structure:

- A textual dimension (AP-Dimension): *treatment proposal*.
- A classical dimension: *anesthesia*.

According to the chart in Figure 8 resulted from OLAP Wonder, the general anesthesia is used with a higher frequency in treatments related to LAPAROTOMY, and the local anesthesia in CYST REMOVAL. According to the medical doctors, this is coherent to real situations since the CYST REMOVAL is less complex intervention, and normally requires local anesthesia, while LAPAROTOMY is more complex and normally, general anesthesia is used.

We can refine the query using a more detailed sentence (apply *Roll-Up*) as *C'={NAILS ENDER OSTEOSYNTHESIS}*. Figure 9 shows the results. As the medical doctor corroborates, the use of NAILS ENDER in OSTEOSYNTHESIS is very specific and when it is applied, general anesthesia is used.

With this example, we can prove the relationships between different matching criteria (*Strong* and *Weak*). We then consider the query *C"={(REMOVAL, CYST)}*, and apply in this case weak matching, comparing with the results for this treatment shown in Figure 8.

9. Conclusions

In this paper, we have presented a new multidimensional model that manages semantical information coming from textual data. The textual dimension in the model lies on a knowledge structure called AP-Structure, which represents more than a simple bag of words. The use of these structures allows the user to query the textual dimension in combination with

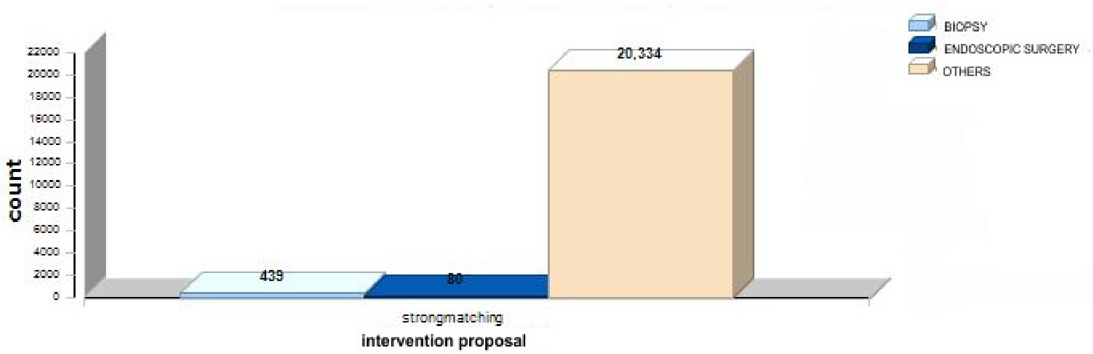

Figure 5: Result for query *C={(SURGERY,ENDOSCOPIC),(BIOPSY)}* (with strong matching)

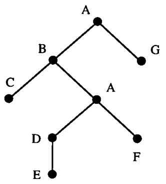

Figure 6: Hierarchy over AP-Dimension *treatment proposal*

the rest non-textual ones. This model enriches the OLAP analysis, so the user can ask the system by means of the traditional operations over a datacube that can contain textual data.

The implementation of the model and the use of the tool in a medical case of study with real data reveals the usefulness of the proposal.

The main future line on work lies on the inclusion of an additional external ontology to complement the query terminology and empower the semantic strength of the proposal.

10. Acknowledgements

This research has been partially funded by Junta de Andalucia projects P07-TIC02786, P10-TIC6109 and P11-TIC7460.

References

[1] M.J. Martin-Bautista, M. Prados, M.A. Vila, and S. Martínez-Folgoso, “A knowledge representation for short texts based on frequent itemsets,” in *Proceedings of International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU’ 2006)*, Paris, France, (2006).

[2] M.J. Martin-Bautista, S. Martínez-Folgoso, and M.A. Vila, “A new semantic representation for short texts,” in *Proceedings of the 10th International Conference on Data Warehousing and Knowledge Discovery (DaWaK’ 2008)*, ser. Lecture Notes in Computer Science (LNCS). Turin, Italy: Springer-Verlag, (2008).

[3] R. Agrawal, A. Gupta, and S. Sarawagi, “Modeling multidimensional databases,” (1995).

[4] R. Kimball, *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*. John Wiley, (1996).

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

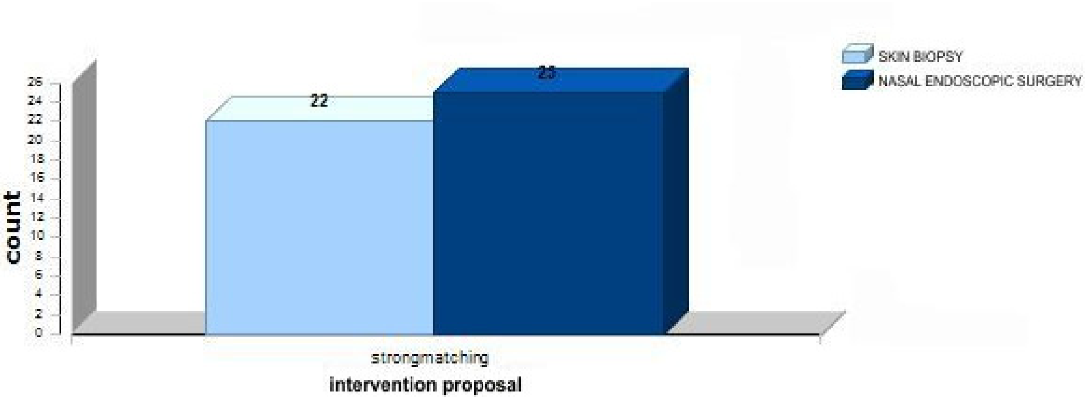

Figure 7: Result for query *{(SURGERY,ENDOSCOPY,NASAL),(BIOPSY,SKIN)}*

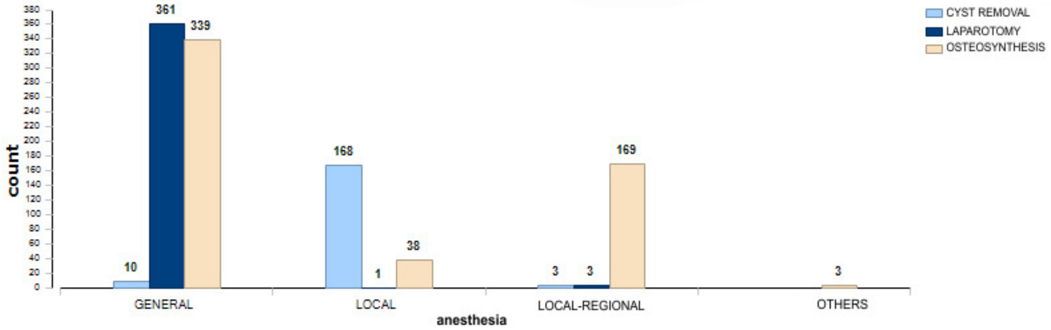

Figure 8: Datacube result for query *C={(LAPAROTOMY),(REMOVAL, CYST) (OSTEOSYNTHESIS)}*

[6] 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** (1999) 289–301.

[7] S. Keith, O. Kaser, and D. Lemire, "Analyzing large collections of electronic text using olap," *The Computing Research Repository (CoRR)*, **abs/cs/0605127** (2006).

[8] M. R. Jensen, T. H. Moller, and T. B. Pedersen, "Specifying olap cubes on xml data," *Journal Of Intelligent Information Systems*, **17** (2001) 200–1.

[9] T. Niemi, M. Ninimaki, J. Nummenmaa, and P. Thanisch, "Applying grid technologies to xml based olap cube construction," in *Proceedings of Desing and Management of Data Warehouses (DMDW' 03) workshop*, Berlin, Germany (2003), 2003–2004.

[10] N. Turkka, J. Kalervo, and N. Timo, "A tool for data cube construction from structurally heterogeneous xml documents," *Journal of the American Society for Information Science and Technology (JASIST)*, **59** (3) (2008) 435–449.

[11] B.-K. Park, H. Han, and I.-Y. Song, "Xml-olap: A multidimensional analysis framework for xml warehouses," in *Proceedings of the 7th International Conference on Data Warehousing and Knowledge Discovery (DaWaK' 2005)*. Copenhagen, Denmark: Springer (2005) 32–42.

[12] F. S. Tseng and A. Y. Chou, "The concept of document warehousing for multi-dimensional

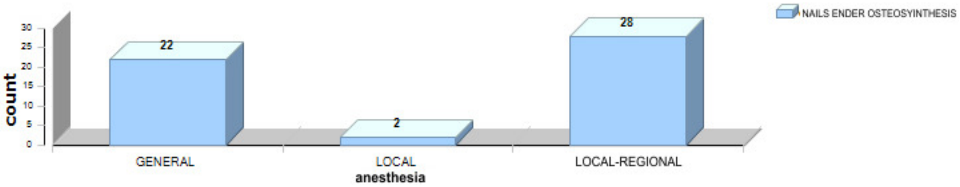

Figure 9: Results for query *C'={NAILS ENDER OSTEOSYNTHESIS}*

modeling of textual-based business intelligence," *Decision Support Systems*, **42** (2) (2006) 727–744.

[13] F. Ravat, O. Teste, R. Tournier, and G. Zurfluh, "Top_keyword: An aggregation function for textual document olap," in *Proceedings of the 10th International Conference on Data Warehousing and Knowledge Discovery (DaWaK' 2008)*, ser. Lecture Notes in Computer Science (LNCS). Turin, Italy: Springer-Verlag (2008) 55–64.

[14] D. Zhang, C. Zhai, and J. Han, "Topic cube: Topic modeling for olap on multidimensional text databases," in *Proceedings of 2009 Society for Industrial and Applied Mathematics (SIAM) International Conference on Data Mining (SDM' 2009)*, Nevada, USA (2009) 1123–1134.

[15] Y. Yu, C. X. Lin, Y. Sun, C. Chen, J. Han, B. Liao, T. Wu, C. Zhai, D. Zhang, and B. Zhao, "inextcube: information network-enhanced text cube," *Proceedings of the Very Large Data Bases (VLDB) Endowment*, **2** (2) (2009) 1622–1625.

[16] C. X. Lin, B. Ding, J. Han, F. Zhu, and B. Zhao, "Text cube: Computing ir measures for multidimensional text database analysis," in *Proceedings of International Conference on Data Mining (ICDM' 2008)*, Pisa, Italy (2008) 905–910.

[17] W. F. Cody, J. T. Kreulen, V. Krishna, and W. S. Spangler, "The integration of business intelligence and knowledge management," *IBM Systems Journal*, **41** (4) (2002) 697–713.

[18] S. Grimes, "New directions for olap," Mar 2006. [Online]. Available: `http://www.intelligententerprise.com/showArticle.jhtml;jsessionid=ICPGHXWQEFX2SQSNDLOSKHSCJUNN2JVN?articleID=181401812` (2006).

[19] A. Vasilakopoulos, M. Bersani, and W. J. Black, "A suite of tools for marking up textual data for temporal text mining scenarios," in *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC' 2004)*, Lisbon, Portugal (2004) 24–30.

[20] F. Ravat, O. Teste, R. Tournier, and G. Zurfluh, "A conceptual model for multidimensional analysis of documents," in *Proceedeings of 26th International Conference on Conceptual Modeling (ER' 2007)*, vol. 4801/2007. Auckland, New Zealand: Springer Berlin / Heidelberg (2007).

[21] J. M. Pérez, R. B. Llavori, and M. J. Aramburu, "A relevance model for a data warehouse contextualized with documents," *Information Processing & Management*, **45** (3) (2009) 356–367.

[22] A. Inokuchi and K. Takeda, "A method for online analytical processing of text data," in *Proceedings of the sixteenth Association Computing Machinery (ACM) conference on information and knowledge Management (CIKM '07)*. New York, NY, USA: Association Computing Machinery (ACM) (2007) 455–464.

[23] M. Banek, A. M. Tjoa, and N. Stolba, "Integrating different grain levels in a medical data

warehouse federation," in *Proceedings of the 8th International Conference on Data Warehousing and Knowledge Discovery (DaWaK' 06)*, Krakow, Poland (2006) 185–194.

[24] P. Torsten and P. Günther, "Ontology-based integration of olap and information retrieval," in *DEXA '03: Proceedings of the 14th International Workshop on Database and Expert Systems Applications*. Washington, DC, USA: IEEE Computer Society, (2003).

[25] N. Marin, M.J. Martin-Bautista, M. Prados, and M.A. Vila, "Enhancing short text retrieval in databases," in *Proceedings of 7th International Conference on Flexible Query Asnwering Systems (FQAS' 2006)*, Milan, Italy, (2006).

[26] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of International Conference on Very Large Data Bases (VLDB' 94)*, Santiago, Chile (1994).

[27] S. Martínez-Folgoso, "Una solución semántica al tratamiento de atributos textuales en un modelo relacional orientado a objetos: Implementación en software libre," Ph.D. dissertation, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, Granada, España (2008).

[28] E. Tejeda-Ávila, "Data warehousing con procesamiento de datos textuales," Master's thesis, Universidad de Granada, Granada, España (2009).

[29] K. G. Batista and E. T. Ávila, "Wonder v 3.0: Servidor olap de libre disposición con soporte a textos libres," Tesis de Pregrado, Universidad de Camagüey, Cuba (2009).

[30] Microsoft-Co., "Directivas de seguridad, administración operativa y comunicaciones," *Microsoft Developer Network (MSDN), Patterns & Practices* [Online]. Available: `http://msdn.microsoft.com/es-es/library/ms978348.aspx`(2006).