# Decisional Architecture for Text Warehousing: ETL-Text Process and Multidimensional Model TWM

Rachid Aknouche
University of Lyon, France
(ERIC, Lyon 2)
rachid.aknouche@univ-lyon2.fr

Ounas Asfari
University of Lyon, France
(ERIC, Lyon 2)
ounas.asfari@univ-lyon2.fr

Fadila Bentayeb
University of Lyon, France
(ERIC, Lyon 2)
fadila.bentayeb@univ-lyon2.fr

Omar Boussaid
University of Lyon, France
(ERIC, Lyon 2)
omar.boussaid@univ-lyon2.fr

## ABSTRACT

In this paper, we propose a decisional architecture composed of three phases. Firstly, a new ETL process for textual data, named ETL-Text, is defined, secondly, a multidimensional text warehouse model, denoted TWM, is designed, and finally, OLAP processing component, named Text-OLAP, is deployed. The proposed TWM model is associated with new dimension types including a meta-data dimension and a semantic dimension. It includes also a new attribute measure based on the language model widely used in information retrieval (IR) area. To validate our approach, we develop a prototype composed of several modules that illustrate the phases of the decisional architecture. Moreover, we use the 20 Newsgroups corpus to carry out our experimentation.

## 1. INTRODUCTION

Data warehousing systems often provide tools to support the decision-making process, especially when data are numerical. Unfortunately, the standard tools are inadequate to deal with the textual data. Thus, it is crucial to develop a new data warehousing and OLAP system to provide the necessary analyzes for the textual data. Towards that end, firstly, an ETL process (Extract-Transform-Load) for the textual data must be established as long as the current techniques of the ETL process are limited for dealing the textual data. The ETL process gathers all the tasks that are necessary to feed and refresh a data warehouse. It is responsible for the extraction of data from several sources, their cleansing, customization and insertion into a data warehouse. Also, current multidimensional models are limited for the analysis of textual data. Thus, some works propose to extend a star or constellation model in order to allow the conceptual modeling of document corpus. In this paper, we propose a new approach for text warehousing process composed of three main components: (1) ETL-Text, (2) Text Warehouse Model (TWM) and (3) Text-OLAP. Here, we are interested on the first two phases; ETL-Text and TWM. Thus, the main contribution is to propose an ETL process for a text data (ETL-Text) and to provide a semantic layer represented in the ETL-Text by concepts extracted based on Wikipedia (http :// www.wikipedia.org) as an external knowledge source. Moreover, the proposed TWM model extends the constellation model to support the representation of textual data in a multidimensional environment based on new dimension types including a meta-data dimension and a semantic dimension.

The rest of this paper is organized as follows. In section 2, we present the related work. Section 3 exposes our decisional architecture, we focus on explaining our proposed ETL-Text and the TWM model. Section 4 overviews the experimental study. Finally, we conclude this paper and we mention the future works in section 5.

## 2. RELATED WORK

Recent studies suggest extending the traditional data warehousing systems to support text data analysis. For instance, in [1]the authors propose a model, named *DocCube*, which offers a new way of access to collections content based on topics, and help the user to improve and formulate his queries. Also, [2] propose a document warehouse for a multidimensional analysis of textual data. They create dimension tables from keywords extracted from the documents. Inspired by the star schema, they use the ID and the number of relevant documents as measures in the *fact* table. In [3], the authors present a multidimensional IR engine (MIRE) that supports the integration of structured data and text. They use the inverted index to process the textual data. In addition, several works, such as [4], propose a multidimensional analysis of documents by extending the classical models or combining the IR techniques with OLAP. Moreover, several works have been developed to extract topics from documents either by using statistical methods, like [5], or by using an external source knowledge to match for each document the most


*The 19th International Conference on Management of Data (COMAD),* 19th-21st Dec, 2013 at Ahmedabad, India.

relevant semantic concepts. In spite of these related work, they are still limited in the extraction of semantics from the textual data. In fact, the data is extracted from various sources and thus it requires different solutions to problems related to heterogeneity models in schema and semantics. The proposed solution here is a new data warehouse model, which allows organizing textual data in a multidimensional environment.

# 3. TEXT WAREHOUSING ARCHITECTURE

In this section, we present a decisional architecture for text warehousing. It covers the construction and the implementation of a data warehouse and adapts new methods to documents analysis. The proposed architecture consists of three phases: (1) ETL-Text phase which can extract, transform and load textual data in a warehouse; (2) Text warehouse modeling TWM phase and (3) Text-Olap phase which allow answering decisional queries using OLAP operators. In this paper, we will focus specifically on the first two phases.

## 3.1 ETL-Text process

The ETL-Text process is one of the essential elements for building a text warehouses. It is responsible for a textual data extraction from various data sources, data cleansing, transforming and loading into a warehouse. The ETL-Text tasks can be summarized as three steps that are: (1)Extract document's textual entities and document's meta-data (2) Transformation of the extracted data (3) Load the resulting data from the transformation step in a warehouse.

### 3.1.1 Extraction and data cleansing phase

Firstly, we parse documents to extract textual entities (terms), this task is called *Tokenization*. Then we filter irrelevant terms for the analysis, known *stopwords*, such as: prepositions, pronouns, some adverbs, etc. They are grouped in a list named *stoplist*. The relevant terms are then assigned to the transformation task to form the so-called, in our approach, the *candidate-terms*.

In order to extract the meta-data from the textual document, $d_n = \{t_1, t_2, ..., t_*\}$ where $t_i \in V$ vocabulary which consists of the index terms, we do not need new techniques to identify and extract them. These operations are already known in a dedicated ETL for handling structured data. Several methods for automatically extracting meta-data are existed in the literature. They are usually based on: regular expressions, rule-based analyzers or machine learning. For instance, we can cite *Metadata Extraction Tool*[1]: developed by the National Library of New Zealand that extracts the meta-data from various types of files such as PDF documents, Microsoft Office documents, image files, etc. Also, we extract the semantic descriptors (concepts). In Wikipedia, for example, these concepts describe the articles subjects and they are represented by the page title. Each concept belongs to at least one category. Our goal, here, is to assign to each textual document in our collection its related concepts and their categories.

### 3.1.2 Transformation and representation

During this phase, we distinguish two transformation types: structural which focuses on the syntactic aspect of textual data and semantic which considers the data semantics.

[1]http://meta-extractor.sourceforge.net/

***Structural Transformation:*** Here, we apply the morph-syntactic process on words that have minor difference in their forms but same meaning, as in the case of the conjugated words. The method used to return these words to their canonical form or *Stem* is called *Stemming*, it consists of the removal of the endings of words. Another solution, known as *lemmatization*, is to use a suffix dictionary to extract the word root with its morphological terms. In our ETL-Text process, for this step, we depend on the Porter algorithm [6] for English language.

***Language Model for documents:*** The *Stems* obtained during the transformation phase will be used to index a document. This step includes the computing of stems weights in the documents by using the language modeling as IR technique [7]. Language model in IR considers a document as a sample language and estimates the probabilities $P(t_i|\theta_{d_n})$ to produce the terms $t_i$ in the language model $\theta_{d_n}$ of the document $d_n$. To weight these stems, we use the Maximum Likelihood Estimation (MLE). It seeks to maximize the data likelihood in a document model and compute the terms probability according to a multinomial distribution defined as follows:

$$P_{ML}(t_i|\theta_d) = \frac{tf_{(t_i,\theta_{d_n})}}{\sum_{t\in\theta_{d_n}} tf_{(t_i,\theta_{d_n})}} \qquad (1)$$

Where $tf_{(t_i,\theta_{d_n})}$ is the occurrence frequency of the term $t_i$ in the document model $\theta_{d_n}$ and $\sum_{t\in\theta_{d_n}} tf_{(t_i,\theta_{d_n})}$ is the total number of occurrence in $\theta_{d_n}$.

**Definition 1** (Candidate-terms). *A candidate-term $\mathcal{T}_{(\mathcal{S}_i,\mathcal{E}_i)}$ of the term $t_i$ in the text document $d_n$ is a tuple $(\mathcal{S}_i, P_{ML}(\mathcal{S}_i|\theta_{d_n}))$ where $\mathcal{S}_i$ denotes the stem of the term $t_i$ in the vocabulary $V$ and $P_{ML}(\mathcal{S}_i|\theta_{d_n})$ is the maximum likelihood estimate MLE of stem $\mathcal{S}_i$ in the language model $\theta_{d_n}$ of the document $d_n$.*

### 3.1.3 Documents enrichment based on Wikipedia

The structural transformation in our ETL-Text process, detailed above, does not capture the semantic relationships between terms. To consider this aspect, we use Wikipedia as an external knowledge source. In Wikipedia, each article describes one topic and is represented by one concept which we denote $A_x$. Also, it has a hierarchical categorization system where each concept belongs to at least one category $C_y$. Our goal, in this operation, is to determine the concepts $A_x$ and their categories $C_y$ that are the most related to each document in our data set.

In order to achieve this goal, we first index the Wikipedia corpus by the Lucene engine (http://lucene.apache.org). Then we use the stems $\mathcal{S}_i$ of each document $d_n = (\mathcal{S}_1, \mathcal{S}_2, ...\mathcal{S}_*)$, obtained during the structural transformation phase, as query terms in the Lucene engine in order to retrieve the relevant concepts to the documents. To compute the similarity between the query and the concepts, we use a metric based on the Kullback-Leibler divergence (KL-divergence). However, the documents are already represented in a vector space of $n$ dimensions by vectors $d_n$=($\mathrm{KL}_{n,1}$,$\mathrm{KL}_{n,2}$, ...$\mathrm{KL}_{n,x}$), where $\mathrm{KL}_{n,x} \in [0,1]$ denotes the KL-divergence score of a document to the concepts $A_x$. These concepts are represented in the same vector space by $A_x = (\mathrm{KL}'_{x,1}, \mathrm{KL}'_{x,2}, \cdots, \mathrm{KL}'_{x,y})$, where $\mathrm{KL}'_{x,y} \in [0,1]$ is also the KL-divergence concepts compared to their Wikipedia categories $C_y$.

Finally, from the two matrices shown in Table 1 and Table

2, we generate the corresponding documents to Wikipedia categories shown in Table 3. This generation is performed by replacing the concepts of the first matrix by their respective categories in the second matrix. Also, the weights of these resulting categories take the KL-divergence score between the concepts and the document.

**Table 1: Documents/Concepts**

| | Concepts Wikipedia | | | |
|---|---|---|---|---|
| **Documents** | $A_1$ | $A_2$ | ... | $A_x$ |
| $d_1$ | $KL_{1,1}$ | $KL_{1,2}$ | ... | $KL_{1,x}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $d_n$ | $KL_{n,1}$ | $KL_{n,2}$ | ... | $KL_{n,x}$ |

**Table 2: Concepts/Categories**

| | Categories Wikipedia | | | |
|---|---|---|---|---|
| **Concepts** | $C_1$ | $C_2$ | ... | $C_y$ |
| $A_1$ | $KL'_{1,1}$ | $KL'_{1,2}$ | ... | $KL'_{1,y}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $A_x$ | $KL'_{x,1}$ | $KL'_{x,2}$ | ... | $KL'_{x,y}$ |

**Table 3: Documents/Categories**

| | Categories Wikipedia | | | |
|---|---|---|---|---|
| **Documents** | $C_1$ | $C_2$ | ... | $C_y$ |
| $d_1$ | $KL''_{1,1}$ | $KL''_{1,2}$ | ... | $KL''_{1,y}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $d_n$ | $KL''_{n,1}$ | $KL''_{n,2}$ | ... | $KL''_{n,y}$ |

## 3.2 TWM: Multidimensional Text Warehouse Model

We propose TWM as an extension of the classical multidimensional model to consider the textual analysis processes. The adopted constellation schema of the TWM model is presented in Figure 1. This schema provides a uniform platform to integrate textual data and to represent their semantics. It organizes text warehouses information into a constellation set of *fact* and *dimension*. In our model, *Fact* represents a textual content of document to be analyzed according to different dimension types. A *dimension* represents the analysis axis according to which we want to observe this fact. Each dimension can be decomposed into several hierarchies constituting different granularity levels. TWM contains beside the classical dimensions, two other types; a meta-data dimension and a semantic dimension in order to represent knowledge that describes documents and their semantics.

### 3.2.1 TWM Dimensions

The dimension can be defined as follows:

**Definition 2** (Dimension). *A dimension $Dim_j$, is defined by a set of attributes $\mathcal{A}t^j_k = \{\mathcal{A}t^j_1, \mathcal{A}t^j_2 \cdots, \mathcal{A}t^j_*\}$, where $\mathcal{A}t^j_k$ is the attribute indexed by $k$ for the dimension indexed by $j$.*

In our TWM model, we distinguish three dimension types:

- ***Classical dimension***: It is characterized by simple attributes which provide descriptive information to specify how the observable data should be summarized. The classical dimension contains data from a single domain. For instance, *Stems* dimension, *document* dimension (see Figure 1).
- ***Meta-data dimension***: This is informational elements describing a document. For example, we can include information explaining a scientific publication, as title, author, language, conference, publisher, etc.
- ***Semantic dimension***: The semantic dimension is represented by a list of concepts which are organized in one hierarchy with several levels. They are related to a textual document and describe its meaning. These concepts are obtained by applying the steps that are previously mentioned in documents enrichment phase of ETL-Text. Thus, it can be defined as:

**Definition 3** (Semantic dimension). *A semantic dimension denoted $Dim^s_j$ is defined by $\{\mathcal{A}t^j_k, \mathcal{H}^s_l\}$ where: $\mathcal{A}t^j_k = \{\mathcal{A}t^j_1, \mathcal{A}t^j_2 \cdots, \mathcal{A}t^j_*\}$ are the attributes of the semantic dimension $Dim^s_j$ and $\mathcal{H}^s_l = \{\mathcal{C}_y, \sqsubseteq\}$ denote the hierarchy of the semantic dimension $Dim^s_j$. This hierarchy is constructed from a set of categories $\mathcal{C}_y = (\mathcal{C}_1 \sqsubseteq \mathcal{C}_2 \sqsubseteq ... \sqsubseteq \mathcal{C}_y)$ derived from Wikipedia and ordered according to a specific order denoted $\sqsubseteq$. These categories represent granularity level $l$ of hierarchy $\mathcal{H}^s_l$.*

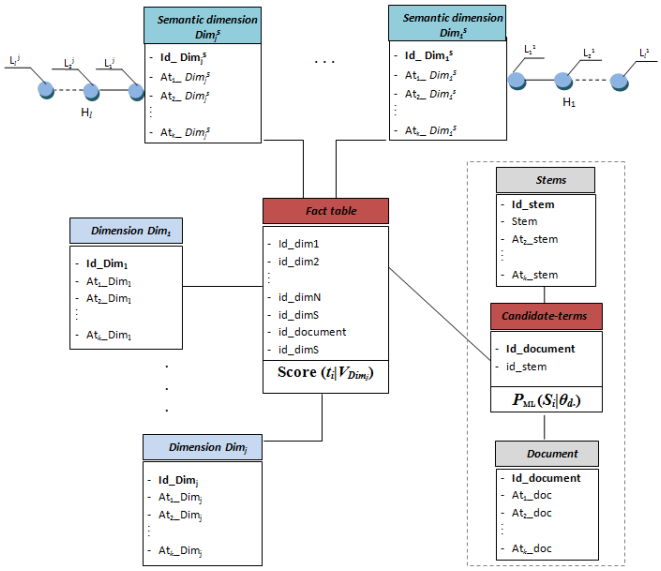


**Figure 1: Constellation text warehouse model**

### 3.2.2 TWM Facts

In our proposed TWM model, the fact is defined as:

**Definition 4** (Fact). *A fact, denoted $\mathcal{F}$, is defined by $\mathcal{F} = \{V^i_{Dim_j}, \mathcal{M}_m\}$ where: $V^i_{Dim_j}$ is a set of attributes' values of dimension $Dim_j$. $\mathcal{M}_m$ is a set of fact measures defined by $\mathcal{M}_m = \{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_*\}$.*

The use of the language modeling has contributed to identify the facts relevant to the analysis. Indeed, they are modeled, in TWM, by two distinct and interconnected constellations. The first links the *Stems* dimension with the documents, whereas the second connects both *"Candidate-terms"* fact table with meta-data and semantic dimensions as shown in Figure 1. The *Stems* dimension includes all terms resulted from the stemming process and thus constituting the lexicon of the document collection. The measure associated to *"Candidate-terms"* is defined by a probabilistic language model. It computes the weight of each value of *Stems* dimension $V^i_{Dim_j}$ in each document language model $\theta_d$. Indeed, we consider each document $d_n$ as a sample of the language

and then we compute the probability of producing the values of dimensions $V_{Dim_j}$ in this document. Also, we adapt the KL-divergence metric to measure the facts of the second constellation. This measure generates a score between probability distributions for different dimensions values and the language model $\theta_q$ of IR query. The advantage of this proposed model is the analyzing and the observing of the textual documents through different dimensions values $V^i_{Dim_j}$ in considering IR queries composed of terms $q = (t_1 t_2..t_*)$.

#### 3.2.3 Language model as a fact measure

We propose to use the probabilistic language modeling as an analysis measure associated to each fact table. A language model is defined as a function which permits estimating the probability of generating term sequences in any modeled language. The language models in information retrieval (IR) are used to compute the probability of generating query $q$ in a document $d$ (i.e. compute: $P(q|D)$); and the documents in the collection $D$ are ranked in descending order of this probability. The proposed measure, in our TWM, permits computing the weight of each dimension value in each document language model denoted $\theta_d$. Here, we consider each document $d_n$ as a language sample and then we estimate the probability of producing the dimensions' values $V^i_{Dim_j}$ in the document. Thus, we adopt the similarity function which is used in IR area between the query model and the document model. Here, the query language model $\theta_q$, is generated from the dimensions' values $V^i_{Dim_j}$. In fact, The dimensions' values constitute the terms $(t_1, t_2, \ldots, t_*)$ of IR query. Then, we calculate the probability of generating these dimension values based on each document model $\theta_{d_n}$ by the following formula:

$$Score(V^i_{Dim_j}, d_n) = \sum_{t \in V} P(t \mid \theta_{V^i_{Dim_j}}) \log P(t \mid \theta_{d_n}) \quad (2)$$

where $\theta_{V^i_{Dim_j}}$ is a language model created for the dimension values $V^i_{Dim_j}$, $P(t \mid \theta_{d_n})$: The probability of term t in the document model $\theta_{d_n}$.

## 4. EXPERIMENTAL EVALUATION

To validate our approach, in particular the proposed ETL-Text process, we developed a software platform for the storage and the analysis of textual data. The modules illustrating the ETL-Text steps have been achieved in Java.

### 4.1 Data Set

***20 Newsgroups*[2]*:*** It is a common benchmark collection of approximately 20,017 documents, partitioned nearly evenly across 20 different newsgroups. The 20 topics are organized into broader categories: computers, recreation, religion, science, for-sale and politics. We used this data set to evaluate the performance of our approach.

***Wikipedia Data:*** In our approach, we depend on Wikipedia data at the semantic enrichment phase. Wikipedia data contents are processed and indexed by using the *Lucene engine*. This is available as dumps download-able from the site (http:// download.wikipedia.org). We used the version of August 2012 which has 4 million articles in English.

### 4.2 ETL-Text phases on the 20 Newsgroups

[2]http ://people.csail.mit.edu/jrennie/20Newsgroups/

We applied the different steps of our ETL-Text process on the 20 Newsgroups corpus, and we evaluated the results of ETL-Text by using the precision/recall metric widely used in the evaluation of IR systems. To calculate these metrics we adopted on the 20 Newsgroups corpus the *Lemur Toolkit* (http//www.lemurproject.com) which is a standard for conducting experiments in IR systems. Also, we used the TREC evaluation tool (the Perl script *ireval.pl* associated with the Lemur Toolkit to interpret the results of the *trec-eval*). The results obtained by applying ETL-Text process on the 20 Newsgroups for the experimental queries show significant improvement in the Precision/Recall compared to those on the same corpus without applying them. Thus, the ETL-Text does not cause any loss of textual data and it loads in the warehouse the most relevant data to the analysis.

## 5. CONCLUSION

We proposed an original approach for building a text warehouse. It uses both natural language processing techniques and information retrieval methods to integrate textual data in a warehouse for analysis purpose. Our contribution consists, first, in a new ETL process appropriate for textual data called ETL-Text. Secondly, we proposed a multidimensional model for a text warehouse called TWM (Text Warehouse Model). TWM is associated with new dimensions types including: a meta-data dimension and a semantic dimension. Also, it has a new analysis measure adapted for text analysis based on the language modeling notion. The documents semantics are extracted by using Wikipedia as an external knowledge source. To validate our approach, we have developed a prototype composed of several processing modules that illustrate the different phases of the ETL-Text. In perspective, we plan to define a new aggregation operators adapted to OLAP analysis on textual data.

## 6. REFERENCES


[1] Mothe, J., Chrisment, C., Dousset, B., Alaux, J.: Doccube: multi-dimensional visualisation and exploration of large document sets. Journal of the American Society for Information Science and Technology, JASIST, Special (54) (2003) 650659

[2] Tseng, F.S.C., Chou, A.Y.H.: The concept of document warehousing for multi-dimensional modeling of textual-based business intelligence. Decis. Support Syst. (42) (November 2006)

[3] McCabe, M.C., Lee, J., Chowdhury, A., Grossman, D., Frieder, O.: On the design and evaluation of a multi-dimensional approach to information retrieval. In: Proceedings of the 23rd annual international ACM SIGIR, New York, NY, USA (2000) 363–365

[4] Zhang, D., Zhai, C., Han, J., Srivastava, A., Oza, N.: Topic modeling for olap on multidimensional text databases: topic cube and its applications. Stat. Anal. Data Min. (2) (December 2009)

[5] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. (2003) 993–1022

[6] Porter, M.F.: Readings in information retrieval. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1997) 313–316

[7] Aknouche, R., Asfari, O., Bentayeb, F., Boussaid, O.: Integrating query context and user context in an information retrieval model based on expanded language modeling. In: CD-ARES. (2012) 244–258