A. Gelbukh, G. Sidorov, A. Guzmán-Arenas. *Relational Data Model in Document Hierarchical Indexing.* In: E. Ranchhold, N. J. Mamede (Eds.) Advances in Natural Language Processing (PorTAL-2002). Lecture Notes in Computer Science, N 2389, Springer-Verlag, 2002, p. 259–262.

# Relational Data Model in Document Hierarchical Indexing*

Alexander Gelbukh, Grigori Sidorov, and Adolfo Guzmán-Arenas

Center for Computing Research, National Polytechnic Institute,
Av. Juan Dios Batiz s/n, Zacatenco 07738, Mexico City, Mexico
{gelbukh, sidorov, aguzman}@cic.ipn.mx

**Abstract.** One of the problems of the development of document indexing and retrieval applications is the usage of hierarchies. In this paper we describe a method of automatic hierarchical indexing using the traditional relational data model. The main idea is to assign continuous numbers to the words (grammatical forms of the words) that characterize the nodes in the hierarchy (concept tree). One of the advantages of the proposed scheme is its simplicity. The system that implements such indexing scheme is described.

## 1 Introduction

Traditional document retrieval systems [1] lack the possibility to perform generalized searches (say, for topic or theme). On the other hand, the systems that permit navigation in a hierarchy, for instance, *Yahoo*, have their own shortcuts, for example:

- Document indexing is usually manual, and, thus, imprecise and subjective,
- Searches with various conditions are not allowed, and
- The documents normally are stored only in one place in the hierarchy though they can deal with several topics in the hierarchy.

These problems can be resolved if we apply automatic indexing of documents using a concept dictionary that usually has a form of a hierarchy of concepts (concept tree), see, for example, [4]. Still, in the paper [4] the task of indexing is not mentioned and the data is obtained from re-processing of the texts each time.

There are several linguistic techniques that allow for more exact and effective document processing (for further document retrieval), see, for example, [2], [3], like morphological analysis or synonymy, etc. Morphological analysis permits to reduce several grammatical forms to only one lexeme (e.g., for lexeme *take* the forms *take*, *took, taken, takes* should be indexed). This may be not of great importance for the languages like English with very few grammatical forms, but it is of great utility for the languages with greater number of grammatical forms for each lexeme as, say, Spanish or Russian.

In this paper we propose the method of the concept tree indexing using the relational data model. The main idea is to give to all grammatical forms that correspond to a lexeme the consecutive numbers, and since the nodes in the tree are

* The work was done under partial support of CONACyT, CGEPI-IPN, and SNI, Mexico.

characterized by a set of lexemes, they also get this numbering. Thus, each grammatical form is represented by the number, while lexemes and nodes are represented by a numeric interval. Usage of the consecutive numbers allows for using of the relational data model that ensures simplicity of the whole method. Namely, it is possible to use any standard database manager available. The corresponding system was developed that implements rather sophisticated query language both for the words and for the nodes of the concept tree.

## 2 Hierarchical Indexing

The documents about *biology* contain words like *plants*, *animals*, *cellules*, etc. Still, if one is interested more specifically in *zoology*, he/she wants to find the document with the words *crocodiles*, or *mammals*, etc. This specification process finishes with individual words. Still, if the user wants to find documents about biology, this also means that the documents about zoology are acceptable, but not vice versa. The structure of the concept tree resembles this specification process: the terminal nodes have the lists of words that correspond to them; the non-terminal nodes unite several terminal or non-terminal nodes (see example below). Each node has only one parent node except the uppermost node that does not have parent.

The suggested method of indexing establishes correspondence between each node (not necessarily the terminal one) and the words that are below the node in such a way that the words are indexed with continuous numbers and each node in the tree is characterized by a numeric interval. This interval is obtained from the words that are below the node.

We have the concept tree in English, because this tree is the language-independent resource. Still, the lists of words that correspond to terminal nodes are language-dependent. In our case, we used Spanish lists, though the list of words in any language can be used.

Technically, the indexation process is as follows: pass all the terminal nodes in the concept tree starting from the leftmost terminal node and enumerate all the words that correspond to them in an order starting from 0. The words in the lists for the terminal nodes are lexemes, so we generate all their grammatical forms and enumerate these forms (not the lexeme itself). At the same time, the terminal nodes are assigned the corresponding numbers, taking the lower bound of the leftmost lexeme and the upper bound of the rightmost lexeme. In case of terminal nodes their children are lexemes.

The next step is to index all non-terminal nodes. The lower value is taken from the leftmost child; the upper value is taken from the corresponding right-most child. In case of non-terminal nodes their children are terminal or non-terminal nodes.

In case that the word belongs to two or more different terminal nodes it is indexed several times with different numbers (with all its grammatical forms).

Having the concept tree indexed, it is possible to index the documents. If the algorithm finds in the documents the grammatical forms that are not in the tree, they are indexed in the same manner as the words in the tree.

After document indexing we have a database that contains numbers of all grammatical forms that were found in the documents. Now it is possible to search rapidly constructing the query. In the query one can use individual grammatical

forms, as well as lexemes, and as well as tree nodes. All of them are substituted by the corresponding intervals of numbers (see Fig. 1 and Fig. 2).

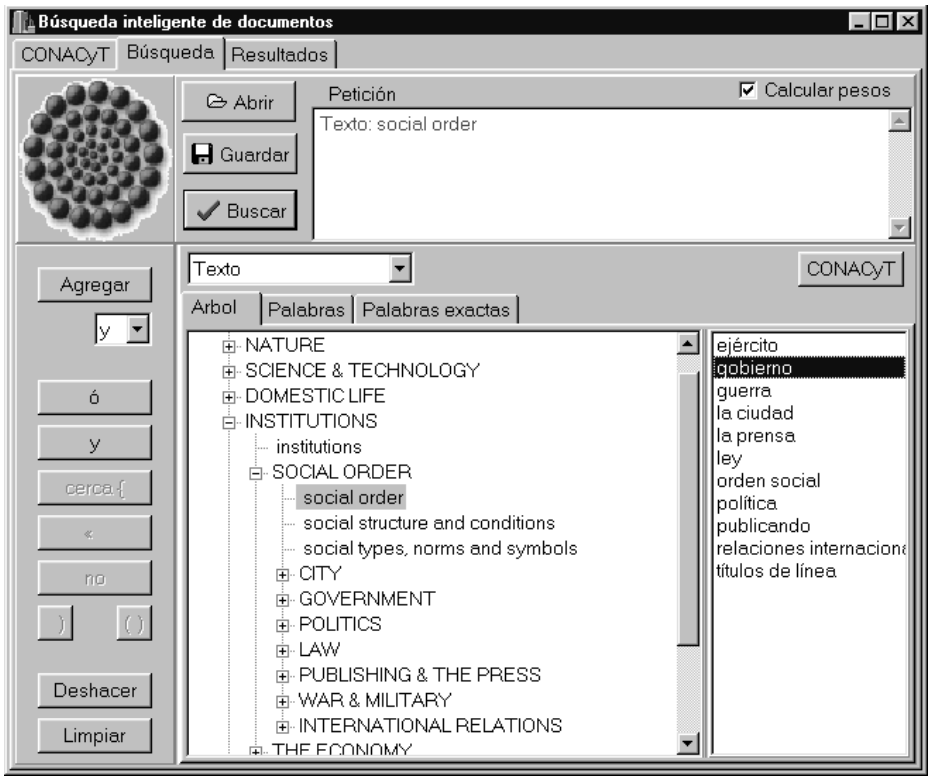


**Fig. 1. Constructing a query using the concept hierarchy.**

## 3 Description of the system

We developed the system that implements the method of indexing described above. The system has sophisticated query language with Boolean operations and proximity criteria. The main advantage of the system is the possibility to search combining themes, lemmas, grammatical forms and their logical combinations.

The query is translated automatically into SQL.

Let us have a look at the example query. Say, the word (lexeme) *cobre* (*brass*) is substituted by four numerical intervals (there are several intervals because the word belongs to several terminal nodes).

The first interval belongs to the terminal node “pure metals” and the second to the terminal node “stones, bricks, tiles, glass and metals”. Their immediate upper node is “MINERALS, METALS & ROCKS”. The third interval belongs to the terminal node “elements” with the upper node “CHEMISTRY”. The fourth interval is the terminal node “money, currency and denominations” with the immediate upper node “THE ECONOMY” (the word *cobre* (*brass*) is used in Spanish to denote a small change).

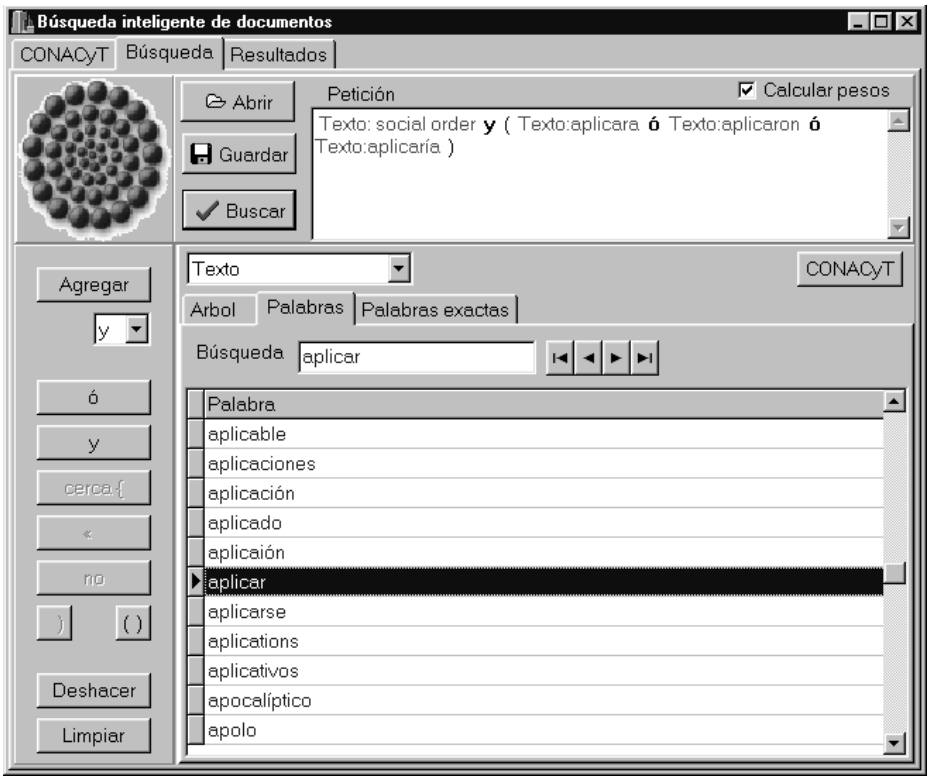


**Fig. 2. Constructing query using grammatical forms and lemmas.**

The buttons on Fig. 1and Fig. 2 that are on the left side of the window permits to insert Boolean operations (*and, or, no*) and brackets. The proximity button permits to search the elements that are "near" (we use the distance of 10 words).

## References


1. Baeza-Yates, R., and B. Ribeiro-Neto. *Modern information retrieval*. Addison-Wesley Longman. 1999.
2. Gelbukh, A. Lazy Query Enrichment: A Simple Method of Indexing Large Specialized Document Bases. Proc. *DEXA-2000*. LNCS 1873, Springer, pp. 526–535.
3. Gelbukh, A. and G. Sidorov. Intelligent system for document retrieval of the Mexican Senate. Proc. *CIC-2000,* IPN, Mexico, pp. 315-321.
4. Gelbukh, A., G. Sidorov, and A. Guzman-Arenas. Use of a weighted topic hierarchy for text retrieval and classification. In *TSD-99*, LNAI 1692, Springer, pp. 130–135.