Alexander Gelbukh, Grigori Sidorov, and Adolfo Guzman-Arenas. *Document indexing with a concept hierarchy*. 1st International Workshop on New Developments in Digital Libraries, NDDL 2001, in conjunction with ICEIS 2001, International Conference on Enterprise Information Systems, Setúbal, Portugal, ICEIS Press, 2001, pp. 47–54.

Document Indexing With a Concept Hierarchy

Alexander Gelbukh, Grigori Sidorov, and Adolfo Guzmán-Arenas

Natural Language Laboratory,
Center for Computing Research (CIC),
National Polytechnic Institute (IPN),
Av. Juan Dios Batiz s/n esq. Mendizabal,
Zacatenco, CP 07738, Mexico D.F., Mexico
{gelbukh, sidorov, aguzman}@cic.ipn.mx

Abstract. We discuss the task of selection of the concepts that describe the contents of a given document. We propose to use a large hierarchical concept dictionary (thesaurus) for this task. A statistical method of document indexing driven by such a dictionary is proposed. The problem of handling non-terminal nodes in the hierarchy is discussed. Common sense-complaint methods of automatically assigning the weights to the nodes and links in the hierarchy are presented. The application of the method in a system Classifier is discussed.

1 Introduction

We consider the task of indexing a document with concepts as mapping the document into the concept dictionary, assigning to each concept in the dictionary a value that reflects its relevance for the given document. Thus, the document is represented by a histogram of its topics. Say, a newspaper article can be about *industry* (60%), *transport* (20%), *science* (10%), etc. Note that these are concepts included in the dictionary rather than the key words directly mentioned in the document; what is more, the document might not contain the word *transport* at all, but instead contain the words *trains*, *railways*, etc.

The problems arising in the compilation and use of a concept hierarchy depend dramatically on its size. In some applications, there is a small set of predefined topics, and a typical document is related to only one topic. For example, this is the case for a governmental reception office where the complaints it receives from the citizens are to be classified to send them to exactly one of the departments of *police*, or *health*, or *environment*, etc.

However, in the case of open texts, such as Internet documents or newspaper articles, the set of possible topics is large and not so well defined, and the majority of the documents are related to several or many topics at the same time. This leads to the necessity of some structuring of the set of topics. The most natural structure for the concepts is a hierarchy. For example, if a document is related to the narrow topics *elections*, *government*, and *party*, then it can be classified as a document on *politics*.

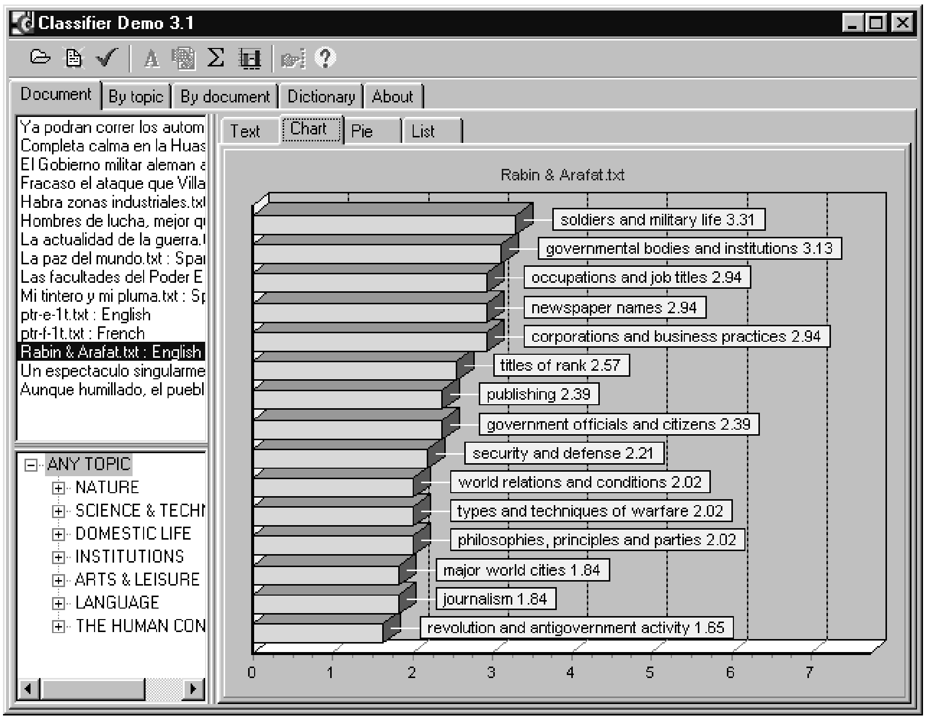

Fig. 1. Topic histogram for a document

Thus, though most of existing dictionary-based systems use “flat” topic dictionaries – keyword groups without any hierarchical structure – in this paper we use a hierarchical dictionary and specifically address the issue of determining the contribution of the top-level concepts. Such a problem does not exist in the “flat” document categorization dictionaries.

We consider the list of topics to be large but fixed, i.e., pre-defined. Our indexing algorithm does not obtain the topics directly from the document body; instead, it relates the document with one of the topics listed in the system dictionary. The result is, thus, the measure (say, in percents) of the corresponding of the document to each of the available topics. Unlike the traditional categorization approach, we consider “fuzzy” categorization, when a document can be indexed with many categories with their corresponding weights. On Fig. 1, a screen shot of our program, CLASSIFIER, is shown with a histogram of the topics of a document.

A problem arises of the optimal, or reasonable, degree of detail for such categorization. For example, when describing the Internet news for an “average” reader, the categories like *animals* or *industry* are quite appropriate, while for the description of articles on zoology such a dictionary would give a trivial answer that all documents are about *animals*. On the other hand, for “average” reader of Internet news it would not be appropriate to categorize the documents by the topics *mammals*, *herptiles*, *crustaceans*, etc., since such a description is too detailed for such a user.

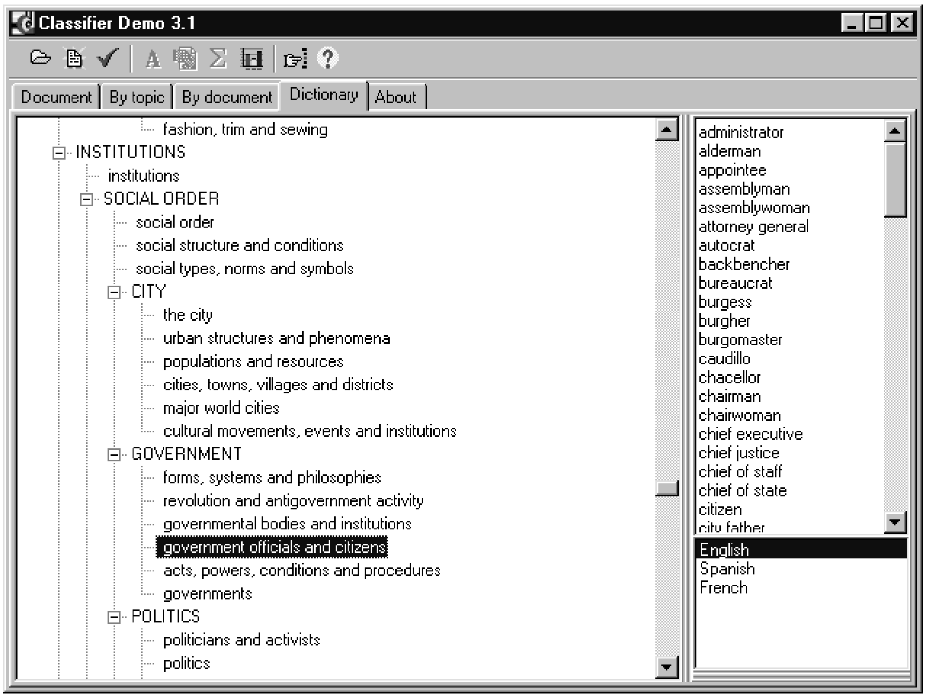

Fig. 2. Hierarchical dictionary used by the system

In this paper, we will discuss the structure of the topic dictionary and the method of the choice and use of topic weights.

2 Concept Hierarchy

In [5] and [6], it was proposed to use a hierarchical dictionary for determining the main themes of a document. Unlike usual methods of indexing, our algorithm does not obtain the candidate topics directly from the body of the document being analyzed. Instead, it relies on a large pre-existing dictionary of topics organized in a tree. Non-terminal nodes of this tree represent major topics, such as *politics* or *nature*. The terminal nodes represent the narrowest topics such as *elections* or *crocodiles*.

Terminal topics are associated with so-called keyword groups. A keyword group is a list of words or expressions related to the situation described by the name of the topic. Such words and expressions are directly used in the text. For example, the topic *religion* can be associated with the words like *church*, *priest*, *candle*, *Bible*, *pray*, *pilgrim*, etc.

Note that these words are connected neither with the headword *religion* nor with each other by any “standard” semantic relation such as subtype, part, actant, etc. This makes compilation of such a dictionary easier than that of a real semantic network dictionary. However, such a dictionary is not a “plain” variant of a semantic network such as WordNet, since some words are grouped together that have no immediate semantic relationship. Thus, such a dictionary cannot be obtained from a semantic network by a trivial transformation.

Fig. 2 shows another example of a dictionary entry. Technically, our CLASSIFIER program manages contact word combinations in the same way as single words.

3 Algorithm

The algorithm of document indexing with the concept thesaurus consists of two parts: individual (leaf) topic detection and propagation of the topics up the tree.

3.1 Topic Detection

The first part of the algorithm is responsible for detection terminal topics, i.e., for answering, individually for each terminal topic, the following question: To what degree this document corresponds to the given topic? In our current implementation, this is done basing on a plain list of words corresponding to the topic. However, in general, a topic can be associated with a procedure. For example, to detect that a document represents an application form relevant to some department of a government office, it may be necessary to analyze the format of the document.

In our implementation, for each keyword group, the number of occurrences of the words corresponding to each (terminal) topic is determined. These numbers are normalized within the document, i.e., divided by the number of words in the document. The accumulated number of occurrences is considered the measure of the correspondence between the document and the topic. Note that the values for this measure of relevance are not normalized since the topics are not mutually exclusive.

3.2 Propagation

The second part of the algorithm propagates the found frequencies up the tree. With this, we can determine that, say, a document mentioning the terminal topics *mammals*, *herptiles*, *crustaceans*, is relevant for the non-terminal topic *animals*, and also *living things*, and also *nature*.

Propagation of the frequencies is crucial for our method. This is necessary to make use of the non-terminal nodes of the hierarchy and to generalize the contents of the document to a degree allowing for its matching with the user's queries that contain more general words than the ones mentioned directly in the document. However, it presents the problem of overgeneralization: applied in the naïve way described here, it always assigns the greatest relevance to the top-level concepts, so that any document is indexed with the concepts *object*, *action*, etc., as its main topics.

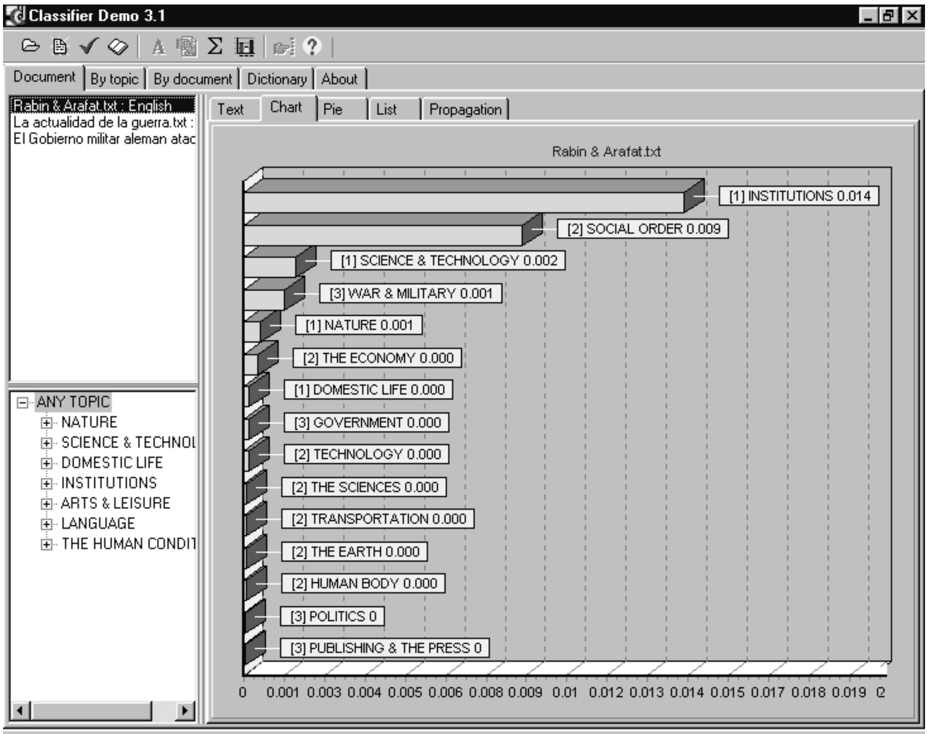

Fig. 3. Non-terminal concepts in the index

The classification algorithm described above is good for answering the question "is this document about *animals*?" but not the question "what about is this document?". Indeed, as we have mentioned, with such an approach taken literally, the answer will be "all the documents in the collection are about *objects* and *actions*," the top nodes of the hierarchy. However, a "reasonable" answer is usually that a document is about *crustaceans*, or *animals*, or *living things*, or *nature*, depending on the user's needs and level, i.e., on the degree of details to which the user is interested in the area.

Thus, we suggest that the answer to the question "what about is this document?" depends on the user. For example, if the document mentions *lobsters*, *shrimps*, *crabs*, and *barnacles*, then for a biologist the answer *crustaceans* would be the best, while for a schoolchild the answer *biology* is better, and for an average newspaper reader, the answer *nature*.

How can we guess this without having to explicitly ask the user? Asking the user about the desired detail degree is not a solution because, first, it is difficult to formulate such a question in an understandable manner, and, second, it is not possible for the user to quantitatively specify the importance of hundreds of topics in the hierarchy. Thus, an automatic way of assigning the importance weights is necessary.

Our hypothesis is that the "universe" of the reader is the base of the documents to which he or she applies the search or classification. In other words, we assume that the reader is a specialist in the contents of the current database being indexed. Thus, the weights of the relevance of topics in our system depend on the current database.

The main requirement to these weights is their discrimination power: an important topic should correspond to a (considerable) subset of documents, while the topics that correspond to nearly all documents in the data base are probably useless, as well as too narrow topics that correspond to few documents in the base.

First we calculate normalized frequencies of each node in the tree for each document. Here w_i^j is the absolute word frequency calculated by summing the values of the adjacent lower nodes and *N* is number of texts in database.

$$r_i^j = \frac{\sum \left(w_i^j\right)}{|N|} \quad (1)$$

This formula is applied recursively starting from the leaf nodes. Now we know for each node for each document its weights r_i^j and can calculate the mean relevance M^j for each tree node *j*.

$$M^j = \frac{\sum_{i \in N} r_i^j}{|N|} \quad (2)$$

Now the weight w^j of a tree node *j* can be estimated as the variation of the relevance r_i^j the topic over the documents of the database. A simple way to measure it is the dispersion. Note that we also have to normalize by *N*:

$$w^j = \frac{\sum_{i \in N} \left(r_i^j - M^j\right)^2}{|N|} \quad (3)$$

So, for the calculation of the resulting weight R_i^j of each node in the hierarchy for the current document we use the combination of the node weight w^j and the relevance weight r_i^j in the current document. Namely:

$$R_i^j = r_i^j \cdot w^j \quad (4)$$

With this approach, for, say, a biological database, the weight of the topics like *animals*, *living things*, *and nature* is low because all the documents equally mention these topics. On the other hand, for newspaper mixture their weight is high.

4 Discussion

We can see two different result of topic detection for the same document on Fig. 1 and Fig. 3. On Fig. 1 we did not use propagation, while Fig. 3 represents propagated weights. Both histograms reflect the document contents but in different ways.

Table 1. Comparison of the first 5 elements with and without propagation

NN	Topics without propagation	Topics with propagation
1	*Solders and military life*	*Institutions*
2	*Government bodies and institutions*	*Social order*
3	*Occupations and job titles*	*Science and technology*
4	*Newspaper names*	*War and military life*
5	*Corporation and business practice*	*Nature / The economy*

Fig. 3 demonstrates the results of application of our method. We used the training database containing over 30 newspaper articles on different themes. Let us compare the five topmost elements in these two histograms.

Having a look at "No propagation" column (from Fig. 1) that reflects the direct contribution of words, we can get an impression which words were the most frequent in the text. Now we can imagine that the corresponding upper topics should be propagated if they were not mentioned equally in all documents of the training database, and, thus have low dispersion.

Let us have a look at "Propagation" column. Note that *Institutions* is the upper node for *Social order*, which is in turn the upper node for *War and military life* to which contributes the terminal node *Solders and military life.* So intuitively it is clear why topic *Institutions* was propagated to the first place taking into account that the terminal node *Government bodies and institutions* also belongs to this topic. It is important to stress that *Any topic* was excluded because it has zero dispersion in the training database. Topic *Science and technology* was propagated to the third place because it has high dispersion (not many documents mention it), and there is rather representative contributing terminal node *Philosophies, principles and parties* (see Fig. 1). It is at the twelfth place. Although this node was rather low without propagation, our method shows that the topic is of interest to a user according to his/her database.

5 Conclusions

We have discussed a method of document indexing driven by a hierarchical concept dictionary. The method is statistical-based and involves the weights of importance of the nodes of the hierarchy for the user. We have suggested that the latter weights depend on the database to which the indexing algorithm is applied. We have discussed the automatic procedure of assigning the corresponding weights to the links and the nodes in the concept hierarchy. The discussed methods have been implemented in a system *Classifier* for document retrieval and investigation of document collections.

Acknowledgements

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

References

1. Chakrabarti, S.; Dom, B.; Agrawal, R.; Raghavan P.: "Using taxonomy, discriminants, and signatures for navigating in text databases", 23rd VLDB Conference, Athenas, Greece.
2. Cohen, W.; Singer, Y. (1996): "Context-sensitive Learning Methods for Text Categorization", Proc. of SIGIR'96.
3. Feldman, Ronen; Dagan, Ido: "Knowledge Discovery in Textual Databases", Knowledge Discovery and Data Mining, (1995), Montreal, Canada.
4. Gelbukh, Alexander; Sidorov, Grigori; Guzmán-Arenas, Adolfo: "A Method of Describing Document Contents through Topic Selection". Proc. of SPIRE'99, International Symposium on String Processing and Information Retrieval, Cancun, Mexico, September 22 – 24. IEEE Computer Society Press, (1999) (http://garota.fismat.umich.mx/spire99), pp. 73-80.
5. Guzmán-arenas, Adolfo: "Finding the main themes in a Spanish document", Journal Expert Systems with Applications, Vol. 14, No. 1/2. (Jan/Feb 1998), pp. 139-148.
6. Guzmán arenas, Adolfo: "Hallando los temas principales en un artículo en español," Soluciones Avanzadas. (1997) Vol. 5, No. 45, p. 58, No. 49, p. 66.
7. Koller, D.; Sahami, M.: "Hierarchically classifying documents using very few words", International Conference on Machine Learning, (1997), 170-178.
8. Ponte, J. M.; Croft W. B.: "Text Segmentation by Topic", First European Conference on Research and Advanced Technology for Digital Libraries, (1997), 113-125.