A Method of Describing Document Contents through Topic Selection

A. Gelbukh, G. Sidorov, and A. Guzmán-Arenas
Natural Language Laboratory, Center for Computing Research (CIC),
National Polytechnic Institute (IPN), Zacatenco 07738, Mexico City, Mexico
{gelbukh,sidorov,aguzman}@pollux.cic.ipn.mx

Abstract

Given a large hierarchical dictionary of concepts, the task of selection of the concepts that describe the contents of a given document is considered. The problem consists in proper handling of the top-level concepts in the hierarchy. As a representation of the document, a histogram of the topics with their respective contribution in the document is used. The contribution is determined by comparison of the document with the "ideal" document for each topic in the dictionary. The "ideal" document for a concept is one that contains only the keywords belonging to this concept, in the proportion to their occurrences in the training corpus. A fast algorithm of comparison for some types of metrics is proposed. The application of the method in a system Classifier *is discussed.*

1. Introduction

We consider the task of representation of the contents of a document by listing its main topics, i.e., the main concepts mentioned in the document. For each topic, we determine its contribution in the document; the representation of the document is a histogram of the concepts with their respective contributions.

Determining the main topics of a document in natural language is important for such applications as document classification, document retrieval [1], text mining [2], investigation of document collections [3], etc.

For example, in document retrieval the documents can be scored by the correspondence of their main topics to the user's request. In text mining, data mining techniques can be applied to discovering trends and deviations of the topics of discussion in the newspapers. In text understanding, topic detection allows selecting the language model [4].

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 by the departments of police, health, 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. This leads to the necessity of some structuring of the set of topics. The most natural structure 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*.

Most of the existing dictionary-based systems use "flat" topic dictionaries – keyword groups without any hierarchical structure. In this paper, we use a hierarchical dictionary and address the issue of determining the contribution of the *top-level* concepts. We introduce the notion of an *ideal* document on a given topic, which supposedly corresponds to the user intuition on what the typical documents on this topic are. We give the formal definition of the ideal document, discuss the method of topic selection based on comparison of the given document with such ideal documents for various topics, and address the issues of computational efficiency of our algorithm.

2. The concept tree and document image

Unlike some other methods of indexing [5], our algorithm does not obtain the candidate topics directly from the body of the document being analyzed. Instead, it relies on a large dictionary ***T*** of topics organized in a tree. Nonterminal nodes of this tree represent major topics, such as *politics* or *nature*. The terminal nodes represent narrower topics such as *elections* or *animals*.

The terminal nodes of the hierarchy are the elementary concepts. They are represented by lists of keywords or keyword combinations that in this hierarchy are considered synonymous. For example, the node *religion* lists the words *church*, *priest*, *candle*, *Bible*, *pray*, *pilgrim*, etc. In our implementation, these keywords can be in different languages, while the concept tree is common for all languages, and the concepts are labeled with English names.

Though the concepts are organized in a tree, a keyword can belong to several concepts. This can be due to either homonymy of the word [6] (e.g., *bill* belongs to *money*, *law*, *tools*, *birds*) or intersection of the topics (e.g., *girl* belongs to *children* and *women*).

In this article, we will suppose that the non-terminal nodes are linked only with other concepts and are not immediately related with any keywords.

Since the words listed in the dictionary under the same terminal concept are considered equivalent for our task, each word of the document stands for one or several (if any) terminal concepts. We represent the document as a vector of numbers of occurrences of each terminal concept. If the word belongs to several concepts, it is counted as one occurrence of each of them. If d_i is the number of occurrences of the terminal concept *i*, then we call the vector $\{d_i\}$ the *image* of the document.

In the rest of the article, we will deal only with such statistical images rather than with documents as sequences of words.

3. Related and previous work

Topic detection for document classification and text segmentation [7] has been the object of extensive research in recent years. A large core of research has been devoted to automatically learning the classification rules using statistical and linguistic methods [8], [9], [10], machine learning methods [11], neural networks, self-organizing maps [12] and connectionist models [13]. In the majority of these studies, the task of automatic construction of the topic hierarchy is considered.

In this article we undertake a different task: given an existing topic hierarchy, to find the concepts that best describe the contents of the document.

In our earlier works [14] and [15], the task of topic selection is understood as the choice of the topic(s) most frequently mentioned in the document. Though in those works a topic hierarchy and the idea of propagating the frequencies up the tree is mentioned, in fact a hierarchy is not used, and the topics are considered as not related to each other in any significant way, i.e., the dictionary is flat, not hierarchical.

In [16], we considered a weighted hierarchy, in which the links between the nodes *i* and *j* have some weights w_j^i: for example, the word *candle* is less relevant for the concept *religion* than the word *pray*. The problem of assignment of such relevance weights was discussed there. For simplicity, in the present paper we will ignore this issue.

In [16] we treated the task of topic detection as discrimination for the purposes of classification, and discussed the corresponding algorithm which relies on the variation of the distribution of the topics over the given document collection. In this article, we consider the task of topic detection in a different aspect, similar to that of abstracting – to give the user an impression of the document contents.

4. Topic selection as document comparison

We will consider the task of topic selection for describing the document contents as answering the following question: Which nodes of the topic hierarchy give the user the best impression of the contents of the given document?

To represent the informal notion of “impression of the contents of the document” we use the notion of the *ideal document image* for the given topic, which we will analyze in more detail below. We suppose that the user has an intuitive notion of the ideal, or the most typical, document on a given topic. When the system labels the document with some node of the concept hierarchy, the user can consider that the contents of the document are approximately equivalent to those of the ideal one for this topic.

Thus, the question “what are the main topics of the document?” can be reformulated as follows: *To the ideal document on which topic is the given document most similar?* In this interpretation, our task is decomposed into two steps:

1. Associate a hypothetical ideal document image with each topic of the hierarchy.

2. Compare the given document with each such ideal document and choose the best match.

In the next sections, we address each of these two issues, providing formal definition of the ideal document and showing some computational advantages of a specific document comparison metric.

5. The ideal document image for a topic

Let us consider what the hypothetical ideal document for the given topic is. One possible way to select the most typical documents could be to choose such documents from a large document collection, on the basis of human expert's opinions. However, we consider this method inappropriate since, first, it involves a great amount of expert handwork and, second, the best methodology for such a procedure is not clear.

Instead, we will artificially construct such documents, basing them only on an unprepared large text corpus with a balanced mixture of topics.

A topic is a node in the concept tree. Since the concept tree is generally an *is-a* hierarchy, each nonterminal node subsumes a subset of the terminal nodes of the tree, so that a topic can be identified with such a subset. We assume the following two hypotheses:

1. The ideal document for a given topic does not contain any keywords that do not belong to this topic.
2. The proportion of the frequencies of the keywords in the ideal document for any topic is the same as in the general text collection (except for the keywords not contained in the document).

The first hypothesis is quite natural: the ideal document on *animals* includes all words related to *animals* but does not include any words related to *computers*. Though in the topic tree, some words can repeat in different topics: say, *mouse* can be both under *animals* and *computers*, we treat them as different nodes, so that only one copy of *mouse* appears in the ideal document on *animals*, while the other does not.

The second hypothesis says that if the word *tail* is twice as frequent as *paw* in the general newspaper mixture, then it will be twice as frequent as *paw* in the texts specifically about *animals*. This statement looks more dubious than the first one and, strictly speaking, does not hold in reality. The following two considerations, though, partially justify it.

First, as is common in the practice of statistics, with a lack of information, the hypothesis of equal distributions is accepted.

A second, more meaningful justification consists in the following. We can consider the general corpus – newspaper mixture that is used to train the system – as consisting of documents, or maybe paragraphs, devoted each one to its own specific topic. Then both words *tail* and *paw* only appear in the paragraphs devoted to *animals*, so that their proportion in the whole corpus is the same as in the topic *animals*. We will ignore here the fact that the frequencies used according to this second hypothesis may be distorted by word ambiguity.

To construct the image of the ideal document for the topic node *N*, we will use a training corpus – a general newspaper mixture. Let ***R*** denote the set of all terminal topics, and k_i the number of occurrences of the terminal topic $i \in \boldsymbol{R}$ in the corpus, i.e., the whole corpus has the document image $\{k_i\}$.

Let ***N*** denote the set of the terminal nodes subordinated to the tree node *N* (then ***R*** is exactly the set of terminals subordinated to the root *R*). Then we define the ideal document for the topic *N* as the one containing only the keywords subordinated to this node. Its image is $\{k_{Ni}\}$, where

$$k_{Ni} = \begin{cases} k_i, & i \in \boldsymbol{N} \\ 0, & i \notin \boldsymbol{N} \end{cases} \qquad (1)$$

Note that the whole corpus image $\{k_i\}$ is equal to $\{k_{Ri}\}$, the ideal document for the root node *R*, i.e., the whole training corpus is the ideal document "about anything."

In the discussion below, the document images will be normalized. However, normalization is impossible for zero documents, e.g., for documents with all $k_i = 0$. There are two possible solutions to this problem. One solution could be to remove such topics from the tree since there is no information about their frequencies. Another possible solution is to a priori add 1 to all frequencies k_i:

$$k_{Ni} = \begin{cases} k_i + 1, & i \in \boldsymbol{N} \\ 0, & i \notin \boldsymbol{N} \end{cases} \qquad (2)$$

The latter method was used in [1] for smoothing the effect of rare events, and we use it too.

6. Comparison metrics

There are many possible ways to measure the distance between two documents *A* and *B* with the images $\{a_i\}$ and $\{b_i\}$. The metrics most frequently used in the literature are the linear (3) and quadratic (4) ones; also a more general metric (5) can be considered:

$$\|A,B\|_1 = \sum_{i\in \boldsymbol{R}} |a_i' - b_i'|, \quad a_i' = \frac{a_i}{\sum_{j\in \boldsymbol{R}} a_j}, \quad b_i' = \frac{b_i}{\sum_{j\in \boldsymbol{R}} b_j} \tag{3}$$

$$\|A,B\|_2 = \sqrt{\sum_{i\in \boldsymbol{R}} (a_i' - b_i')^2}, \ a_i' = \frac{a_i}{\sqrt{\sum_{j\in \boldsymbol{R}} a_j^2}}, \ b_i' = \frac{b_i}{\sqrt{\sum_{j\in \boldsymbol{R}} b_j^2}} \tag{4}$$

$$\|A,B\|_n = \sqrt[n]{\sum_{i\in \boldsymbol{R}} |a_i' - b_i'|^n}, \ a_i' = \frac{a_i}{\sqrt[n]{\sum_{j\in \boldsymbol{R}} a_j^n}}, \ b_i' = \frac{b_i}{\sqrt[n]{\sum_{j\in \boldsymbol{R}} b_j^n}} \tag{5}$$

Other metrics are also proposed in the literature. For example, in [2] an information-based asymmetric distance measure is used:

$$\|A,B\|_{\text{info}} = \sum_{i\in \boldsymbol{R}} a_i' \log \frac{a_i'}{b_i'}, \quad a_i' = \frac{a_i}{\sum_{j\in \boldsymbol{R}} a_j}, \quad b_i' = \frac{b_i}{\sum_{j\in \boldsymbol{R}} b_j} \tag{6}$$

though such a measure is not adequate for our purposes since it is not defined for $k_i = 0$.

The issue of justification and meaningful choice between such metrics is complex and we do not discuss it here in detail, though we provide some discussion in section 9.

Note that since we use a normalized document image $\{d_i'\}$, our method can not work with empty documents or with documents that do not contain any keywords.

7. Computational efficiency

In many works the quadratic metric (4) is preferred because of computational advantages it provides. We will also show that it significantly speeds up the calculation process, as well as some other metrics of the type (5).

With an arbitrary metric like (3), the algorithm described in the section 4 has the complexity $|\boldsymbol{T}| \times |\boldsymbol{R}|$, where $|\boldsymbol{T}|$ is the total number of nodes in the tree $\boldsymbol{T}$, and $|\boldsymbol{R}|$ is the total number of terminal nodes, i.e., the dimension of the document image. Indeed, in each of $|\boldsymbol{T}|$ nodes of $\boldsymbol{T}$ the algorithm requires calculation of the expression like (3), which in its turn requires $|\boldsymbol{R}|$ elementary operations. We do not consider the operations required for calculating d_i' and k_i' since they are calculated once for the document and at the stage of training the model, respectively.

We will show that with the quadratic metric, the computational complexity of the algorithm from the section 4 can be reduced to the order of $|\boldsymbol{R}| + |\boldsymbol{T}|$. Let us rewrite the expression (4) as

$$\begin{aligned}\|A,B\|_2^2 &= \sum_{i\in \boldsymbol{R}} (a_i' - b_i')^2 \\ &= \sum_{i\in \boldsymbol{R}} a_i'^2 + \sum_{i\in \boldsymbol{R}} b_i'^2 - 2\sum_{i\in \boldsymbol{R}} a_i' b_i' \\ &= 2 - 2\frac{1}{\sum_{j\in \boldsymbol{R}} a_j^2}\frac{1}{\sum_{j\in \boldsymbol{R}} b_j^2}\sum_{i\in \boldsymbol{R}} a_i b_i\end{aligned} \tag{7}$$

Let K_N be the ideal document for the topic node N, and $\{k_{N\,i}\}$ be its image. Then, because of the zeroes in (1), the summation in the latter expression can be performed only by the set $\boldsymbol{N}$ of the terminal modes subordinated to N and not by the entire $\boldsymbol{R}$:

$$\|K_N, D\|_2^2 = 2 - 2\frac{1}{\sum_{j\in \boldsymbol{N}} k_j^2}\frac{1}{\sum_{j\in \boldsymbol{R}} d_j^2}\sum_{i\in \boldsymbol{N}} k_i d_i \tag{8}$$

Denoting the corresponding parts of the latter formula by $F(N)$, G, and $w(i)$, we can rewrite this expression as

$$\|K_N, D\|_2 = \sqrt{2 - 2F(N)G\sum_{i\in \boldsymbol{N}} w(i)} \tag{9}$$

where the coefficient $F(N)$ depends only on the node N and does not depend on the document D, while the coefficients G and $w(i)$ depend on the document D but do not depend on the node N.

Let us consider the number of operations required during runtime to calculate (9). The value $F(N)$ can be pre-calculated at the time of training the model and thus is not calculated in runtime at all. Calculation of G for a given document requires on the order of $|\boldsymbol{R}|$ operations.

At the first glance, calculating all $W(N) = \sum_{i\in \boldsymbol{N}} w(i)$ seems to require on the order of $|\boldsymbol{T}| \times |\boldsymbol{R}|$ operations. In fact, due to linearity of this expression in N, it can be recursively calculated for all nodes in only $|\boldsymbol{T}| - 1$ steps. For the terminal nodes there is nothing to calculate. For any non-terminal node N, $W(N) = \sum_{N_s \leftarrow N} W(N_s)$ with the summation only by the nodes N_s immediately subordinated to N, $N_s \leftarrow N$. This results in the number of additions equal to the number of the arcs in the tree, $|\boldsymbol{T}| - 1$. This algorithm is discussed in the section 8.

Thus, the main algorithm from section 4 with the quadratic metric (4) requires only on the order of $|\boldsymbol{T}| + |\boldsymbol{R}|$ operations. The memory requirements are of the order of $|\boldsymbol{T}|$, including both the data learned at the time of training the model, which are kept in the system, and the data related to the document being analyzed.

Note that this result can be generalized to an arbitrary metric (5) with any even $n \geq 2$. Similarly to (7), the expression (5) can be rewritten as

$$\begin{aligned} \left\| A, B \right\|_n^n &= \sum_{i \in \boldsymbol{R}} \left(a_i' - b_i' \right)^n \\ &= \sum_{i \in \boldsymbol{R}} \sum_{m=0}^{n} (-1)^{n-m} \begin{pmatrix} n \\ m \end{pmatrix} a_i'^{\,m} b_i'^{\,(n-m)} \qquad (10) \\ &= 2 + \sum_{m=1}^{n-1} \frac{(-1)^{n-m} \begin{pmatrix} n \\ m \end{pmatrix}}{\left(\sum_{j \in \boldsymbol{R}} a_j^n \right)^{\frac{m}{n}} \left(\sum_{j \in \boldsymbol{R}} b_j^n \right)^{\frac{n-m}{n}}} \sum_{i \in \boldsymbol{R}} a_i^m b_i^{n-m} \end{aligned}$$

which in turn for the ideal document image $k_i = 0$ due to the zeroes in (1) can be rewritten as

$$\left\| K_N, D \right\|_n^n = 2 + \sum_{m=1}^{n-1} \frac{(-1)^{(n-m)} \binom{n}{m}}{\left(\sum_{j \in \boldsymbol{N}} k_j^n \right)^{\frac{m}{n}} \left(\sum_{j \in \boldsymbol{R}} d_j^n \right)^{\frac{n-m}{n}}} \sum_{i \in \boldsymbol{N}} k_i^m d_i^{n-m} \qquad (11)$$

with summation only by $\boldsymbol{N}$, and thus

$$\left\| K_N, D \right\|_n = \sqrt[n]{2 + \sum_{m=1}^{n-1} Q_m F_m(N) G_m \sum_{i \in \boldsymbol{N}} w_m(i)} \qquad (12)$$

where only a fixed number of values $F_m(N)$ depend on the node N, while they do not depend on the document D and thus can be compiled at the time of training the model.

All our algorithms can be reformulated for this case, with the complexity of order $n\,(|\boldsymbol{T}| + |\boldsymbol{R}|)$ and memory requirements of order $n\,|\boldsymbol{T}|$, though we will not give such a general form here.

8. Algorithm

We will discuss only the quadratic metric (4), though a similar algorithm can be built for an arbitrary metric (5).

The algorithm corresponding to the discussion in section 4 uses the following variables: the array $F(N)$ numbered by the nodes of the tree T, a simple variable G depending on the document D being analyzed, and two arrays k_i and d_i numbered by the terminal nodes.

The algorithm uses a recursive procedure to promote a value up the tree, i.e., to calculate in each node N the sum $X(N) = \sum_{i \in N} x(i)$ of some values $x(i)$, with summation by all terminal nodes subordinated to N. This procedure has been discussed in section 7 and has a complexity of order $|\boldsymbol{T}| - 1$. It is applied to the root node of the tree and consists in the following conditions:

1. If N is a terminal node, set $X(N) = x(i)$ with the appropriate number i.
2. If N is a non-terminal node, apply the same procedure to each node N_s immediately subordinated to N, $N_s \leftarrow N$, and set $X(N) = \sum_{N_s \leftarrow N} X(N_s)$; here the summation is done only by the nodes immediately subordinated to N.

In the following sections we will apply this procedure to different expressions $x(i)$, obtaining different functions $X(N)$.

Also, the algorithm uses the following procedure for building a document image $\{a_i\}$ given a document A:

1. Set all $a_i = 0$.
2. For each word of the document, if it belongs to the list of keywords for some terminal node i, increment a_i by 1. A word can belong to more than one terminal node; in this case all corresponding a_i are incremented.

The complexity of this procedure is of the order $|A| \log |L|$, where $|A|$ is the total number of words in the document A and $|L|$ is the total number of the keywords in the system lexicon.

8.1. Training the model

The data learned from the training corpus and kept in the system are $F(N)$ and $\{k_i\}$. The total size of the data is $|\boldsymbol{T}| + |\boldsymbol{R}|$ (it can be reduced to $|\boldsymbol{T}|$ by storing $F(N)$ only for non-terminal nodes), where $|\boldsymbol{R}|$ is the number of terminal nodes and $|\boldsymbol{T}|$ of all nodes in the tree.

The input for the training algorithm is a large corpus of general newspaper mixture. The algorithm works as follows:

1. Apply the procedure of building the document image to the training corpus to build $\{k_i\}$. The complexity of this step is of the order $|K| \log |L|$, where $|K|$ is the number of words in the training corpus.
2. Calculate all values of $F(N) = 1 \big/ \sum_{i \in \boldsymbol{N}} k_i^2$ by first applying the promotion procedure to the expression

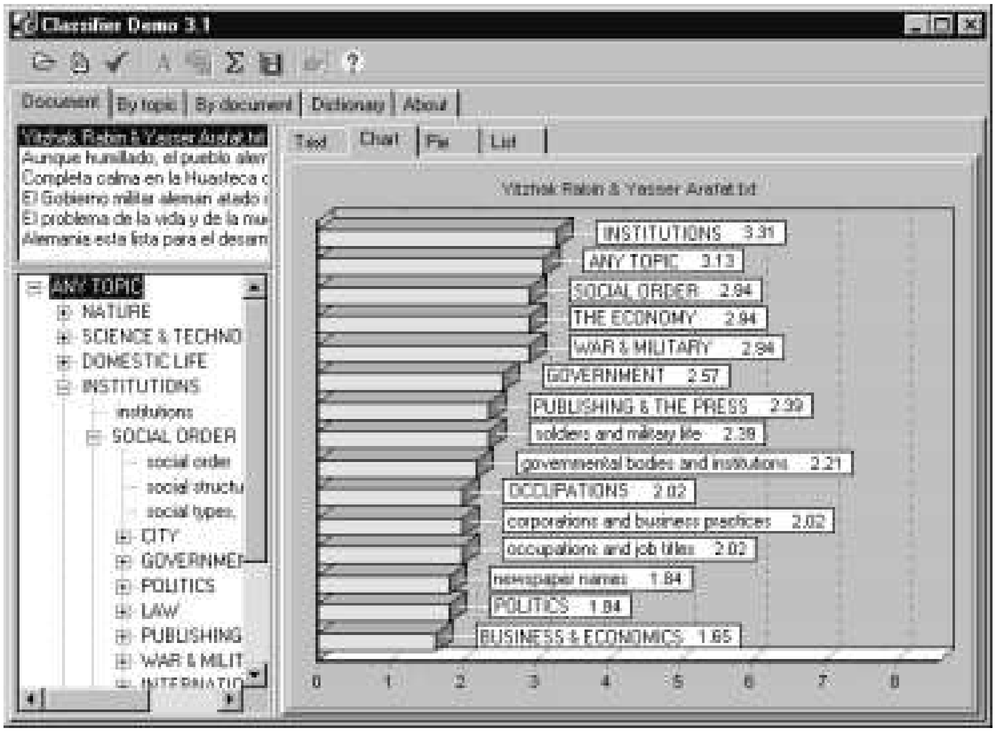
Classifier Demo 3.1
Document
By topic
By document
Dictionary
About
Yitzhak Rabin & Yasser Arafat.txt
Aunque humillado, el pueblo alem
Completa calma en la Huasteca c
El Gobierno militar alemán atado
El problema de la vida y de la mu
Alemania esta lista para el desarr
ANY TOPIC
NATURE
SCIENCE & TECHNO
DOMESTIC LIFE
INSTITUTIONS
institutions
SOCIAL ORDER
social order
social structu
social types.
CITY
GOVERNMEN
POLITICS
LAW
PUBLISHING
WAR & MILIT
INTERNATIO
Text
Chart
Pie
List
Yitzhak Rabin & Yasser Arafat.txt
INSTITUTIONS 3.31
ANY TOPIC 3.13
SOCIAL ORDER 2.94
THE ECONOMY 2.94
WAR & MILITARY 2.94
GOVERNMENT 2.57
PUBLISHING & THE PRESS 2.39
soldiers and military life 2.39
governmental bodies and institutions 2.21
OCCUPATIONS 2.02
corporations and business practices 2.02
occupations and job titles 2.02
newspaper names 1.84
POLITICS 1.84
BUSINESS & ECONOMICS 1.65
0
1
2
3
4
5
6
7
8

Figure 1. A screenshot of the Classifyer system: the topics of the article *Yitzhak Rabin and Yasser Arafat.*

$x(i) = k_i^2$ and after that inverting the resulting value in each node. The complexity of this step is of the order $|\boldsymbol{T}| - 1$.

The total complexity of training the model is of the order $|K| \log |L| + |\boldsymbol{T}| - 1$.

8.2. Working with the document

In runtime, i.e., when working with a given document, the algorithm computes and uses the values G and $\{d_i\}$. The total size of the data for a given document is $|\boldsymbol{R}| + 1$, where $|\boldsymbol{R}|$ is the number of terminal nodes. Given a document D, the algorithm works as follows:

1. Apply the procedure of building the document image to the document to build $\{d_i\}$. The complexity of this step is of the order $|D| \log |L|$.
2. Calculate the value of $G = 1 \big/ \sum_{i \in \boldsymbol{R}} d_i^2$. The complexity is of the order $|\boldsymbol{R}|$.
3. Calculate all values $\|K_N, D\|_2$. For this, first apply the promotion procedure described above to the expression $x(i) = w(i) = k_i d_i$ and after that calculate the expression (9) for each node N. The complexity is of the order $|\boldsymbol{T}| - 1$.
4. Choose the node with the least value of $\|K_N, D\|_2$, or arrange the nodes in the order of increasing of this value. In the latter case, the complexity is of the order $|\boldsymbol{T}| \log |\boldsymbol{T}|$.

The total complexity is of the order $2|\boldsymbol{R}| + |\boldsymbol{T}| - 1$ if only the best topic is to be found. However, if entire histogram of the topics ordered from the best to the worst one is to be built, additional $|\boldsymbol{T}| \log |\boldsymbol{T}|$ operations are necessary.

9. Implementation and experimental results

The algorithm was implemented in a multifunction system *Classifier*. The system allows the user to view the histogram of the topics expressed in a given document, see Figure 1.

The system also allows the user to retrieve from the data base the documents corresponding to the given topic, or to view a histogram of the documents in the data base that correspond to the given topic, ordered from the best to the worst. Among the functions of the *Classifier* system is the ability to compare documents using the topic tree and search for the documents which are the most similar to the given one.

Our experiments were conducted with Spanish documents. As the training corpus, we used the publicly available Spanish corpus LEXESP provided to us by the Polytechnic University of Catalonia (UPC), Spain. The corpus contains approximately 3 million words.

As the concept tree, we used the multilingual data from the *Clasitex* system [15], with the concepts with English labels, and with English, Spanish, and French vocabulary. The tree consists of 796 nodes, of which 607 nodes are terminal. The English vocabulary consists of 37946 keywords or keyword combinations (like *hot dog*).

The results were assessed qualitatively, based on the opinion of human experts. For each document, two tests were carried out.

In the first test, the expected main topics were assigned to the documents manually, and the system results were compared with these *a priori* expert judgments. In more than 90% of the cases, the expected main topic was within the first 10 topics (of total 796) in the histogram.

In the second test, the top 10 topics reported as the best by the program were estimated by the human testers for their intuitive appropriateness for the document. In more than 80% of the cases, the testers estimated the results as acceptable.

We have experimented with different metrics, such as (3) to (5). The metrics exhibited rather similar behavior, though the metrics of the type (5) with higher *n* tended to give slightly higher priority to lower nodes of the tree (like *elections*), while the linear metric (3) emphasized more general topics (like *politics*). We also tried the expression

$$\|A, B\|_{2,1}^{2} = 1 - a_i' b_i', \quad a_i' = \frac{a_i}{\sum\limits_{j \in \boldsymbol{R}} a_j}, \quad b_i' = \frac{b_i}{\sum\limits_{j \in \boldsymbol{R}} b_j} \qquad (13)$$

which tended to over-emphasize the terminal nodes, i.e., the most narrow topics. In our opinion, the quadratic metric provides a good compromise between the quality, simplicity, and computational efficiency.

As an example, let us consider the results for a newspaper article *Yitzhak Rabin and Yasser Arafat*, which was evaluated by the human expert as related to *politics*, *war*, *social institutions*. The following table shows the system output, with the distances measured by the quadratic metric $\|K_N, D\|_2$ (4). Terminal nodes are given in lowercase letters, non-terminal in capitals:

Rank	Distance	Topic
1	0.725	*INSTITUTIONS*
2	0.778	*ANY TOPIC*
3	0.855	*SOCIAL ORDER*
4	1.114	*THE ECONOMY*
5	1.137	*WAR, MILITARY*
6	1.160	*GOVERNMENT*
7	1.182	*PUBLISHING, THE PRESS*
8	1.199	*soldiers, military life*
9	1.212	*governmental bodies, institutions*
10	1.213	*OCCUPATIONS*
11	1.225	*corporations, business practices*
12	1.225	*occupations, job titles*
13	1.225	*newspaper names*
14	1.225	*POLITICS*
15	1.237	*BUSINESS & ECONOMICS*

Note that in the dictionary by *institutions*, social institutions such as government or politics are meant.

Here is a fragment of the topic tree with the ranks corresponding to the previous table. This table illustrates the pattern of topic ranking. The topics nearest in the tree to the main detected topic, in this case *institutions*, have the best ranking.

Topic subtree	Rank
ANY TOPIC	2
⎣ *INSTITUTIONS*	1
⎣ *SOCIAL ORDER*	3
│ ⎣ *GOVERNMENT*	6
│ │ ⎣ *governmental bodies and institutions*	9
│ ⎣ *POLITICS*	14
│ ⎣ *PUBLISHING & THE PRESS*	7
│ │ ⎣ *newspaper names*	13
│ ⎣ *WAR & MILITARY*	5
│ ⎣ *soldiers and military life*	8
⎣ *THE ECONOMY*	4
⎣ *BUSINESS & ECONOMICS*	15
│ ⎣ *corporations and business practices*	11
⎣ *OCCUPATIONS*	10

Note that the root topic – *any topic* – obtained a high rank, which is not desirable. In our future work we plan to investigate other metrics and address the issue of the choice of the optimal one.

10. Conclusions and future work

As the method of concept selection for representation of the contents of a document, comparison with an "ideal" document for each of the topics available in the dictionary was suggested. The method of automatic construction of such an ideal document for a topic was proposed. The issue of choice of the metric for the comparison was discussed. For some types of frequently used metrics, a faster algorithm of calculation was described.

The method has been implemented in a system *Classifier* for document retrieval and investigation of document collections. In the experiments, different metrics exhibited slightly different behavior. As the experiments have shown, with the quadratic metric, the ranking of the root node is too high. In future work, the issue of choice of the best metric is to be addressed.

11. Acknowledgments

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

12. References

[1] S. Chakrabarti, B. Dom, R. Agrawal, P. Raghavan. "Using taxonomy, discriminants, and signatures for navigating in text databases," *23rd VLDB Conference*, Athenas, Greece, 1997.

[2] R. Feldman and Ido Dagan. "Knowledge Discovery in Textual Databases." *Knowledge Discovery and Data Mining*, Montreal, Canada, 1995.

[3] John Light. "A distributed, graphical, topic-oriented document search system." *CIKM '97, Proceedings of the sixth international conference on Information and knowledge management*, pp. 285-292

[4] K. Seymore and R. Rosenfeld. "Using story topics for language model adaptation," *Proc. of Eurospeech'97*, 1997.

[5] Yoshiki Niwa, Shingo Nishioka, Makoto Iwayama, Akihiko Takano, Yoshihiko Nitta. "Topic Graph Generation for Query Navigation: Use of Frequency Classes for Topic Extraction," *NLPRS'97, Natural Language Processing Pacific Rim Symposium '97*, Phuket, Thailand, Dec. 1997, pp. 95-100.

[6] Krowetz, B. "Homonymy and Polysemy in Information Retrieval," *35th Annual Meeting of the Association for Computational Linguistics*, 1997, pp. 72-79

[7] J.M. Ponte and W. B. Croft. "Text Segmentation by Topic," *First European Conference on Research and Advanced Technology for Digital Libraries*, 1997, pp. 113-125.

[8] Chidanand Apté, Fred Damerau, and Sholom M. Weiss. "Automated learning of decision rules for text categorization." *ACM Transactions on Information Systems*. Vol. 12, No. 3 (July 1994), pp. 233-251.

[9] Krishna Bharat and Monika Henzinger. "Improved algorithms for topic distillation in hyper-linked environments," *21st International ACM SIGIR Conference*, 1998.

[10] Cohen, W., Singer, Y. "Context-sensitive Learning Methods for Text Categorization," *Proc. of SIGIR'96*, 1996.

[11] Koller, D. and Sahami, M. "Hierarchically classifying documents using very few words." *International Conference on Machine Learning*, 1997, pp. 170-178.

[12] Heikki Hyötyniemi. "Text Document Classification with Self-Organizing Maps," *STeP'96, Genes, Nets and Symbols*, Alander, J., Honkela, T., and Jakobsson, M. (eds.), Finnish Artificial Intelligence Society, pp. 64-72.

[13] D.X. Le, G. Thoma, H. Weschler. "Document Classification using Connectionist Models." *IEEE International Conference on Neural Networks*, Orlando, FL, June 28 – July 2, 1994 Vol. 5, pp. 3009-3014.

[14] Adolfo Guzmán Arenas. "Hallando los temas principales en un artículo en español," *Soluciones Avanzadas*. Vol. 5, No. 45, p. 58, No. 49, 1997, p. 66.

[15] Adolfo Guzmán-Arenas. "Finding the main themes in a Spanish document," *Journal Expert Systems with Applications*, Vol. 14, No. 1/2. Jan/Feb 1998, pp. 139-148.

[16] A. Gelbukh, G. Sidorov, and A. Guzmán-Arenas. "Use of a Weighted Topic Hierarchy for Document Classification," *TSD-99, Text, Speech, Dialogue*. Prague, September 1999, to appear.