Use of a Weighted Topic Hierarchy for Document Classification*

Alexander Gelbukh, Grigori Sidorov, and Adolfo Guzman-Arénas

Natural Language Laboratory,
Center for Computing Research (CIC), National Polytechnic Institute (IPN),
Av. Juan de Dios Bátiz, CP 07738, Zacatenco, Mexico City, Mexico
{gelbukh, sidorov, aguzman}@pollux.cic.ipn.mx

Abstract. A statistical method of document classification driven by a hierarchical topic dictionary is proposed. The method uses a dictionary with a simple structure and is insensible to inaccuracies in the dictionary. Two kinds of weights of dictionary entries, namely, relevance and discrimination weights are discussed. The first type of weights is associated with the links between words and topics and between the nodes in the tree, while the weights of the second type depend on user database. A common sense-complaint way of assignment of these weights to the topics is presented. A system for text classification *Classifier* based on the discussed method is described.

1 Introduction

We consider the task of classification by their topics: for example, some documents are about *animals*, and some about *industry*. This task is important in information retrieval, classification of document flows, such as incoming documents in a large government office, filtration of document flows, such as Internet news, and in many other applications. In recent years appeared many articles on the theme, see, for example, [1]-[3], [7]-[10].

In this paper we consider the list of topics to be large but fixed. Our algorithm does not obtain the topics 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.

A problem arises of the optimal, or reasonable, degree of detail for such classification. For example, when classifying the Internet news for an "average" reader, the categories like *animals* or *industry* are quite appropriate, while for classification 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 classify the documents by the topics *mammals, herptiles, crustaceans*, etc.

* The work partially supported by DEPI-IPN, CONACyT (26424-A), and REDII, Mexico.

2 Topic Hierarchy

In [5] and [6], it was proposed to use a hierarchical dictionary for determining the main themes of a document. Technically, the dictionary consists of two parts: *keyword groups* and a *hierarchy* of such topics.

A keyword group is a list of words or expressions related to the situation described by the name of the topic. For example, the topic *religion* could list the words like *church, priest, candle, Bible, pray, pilgrim*, etc. Technically, our *Classifier* program manages word combinations in the same way as single words.

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 much easier than 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.

The other part of the dictionary is the topic tree, which organizes the topics, as integral units, into a hierarchy or, more generally, a lattice (since some topics can belong to several nodes of the hierarchy).

3 Basic Classification Algorithm

The algorithm of application of the dictionary to the task of topic detection also consists of two parts: individual (leaf) topic detection and propagation of the topics up the tree. *The first part* of the algorithm is responsible for detection individual (leaf) topics, i.e., for answering, topic by topic, the question: to what degree this document corresponds to the given topic? Such a question is answered for each topic individually. We call the element that answers such a question for a fixed topic a voter[2]. In our current implementation, a voter is based on a plain list of words corresponding to the topic; however, in general a voter can be associated with a procedure: for example, to detect that a document is an application form relevant to some department of a government office. Then it may be necessary to analyze the format of the document.

In our current system, for each keyword group the number of occurrences of the words corresponding to each (leaf) 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 to be the measure of correspondence of the document to the topic. Note that the values for this measure of relevance are not normalized since the topics are not mutually exclusive.

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

[2] The terms *tester* or *topic agent* could be also appropriate.

Instead of simple lists of words, some numeric weights can be used by the algorithm to define the quantitative measures of relevance of the words for topics and the measure of importance of the nodes of the hierarchy. Thus, there are two kind of such weights: the weights of links in the hierarchy and the weights associated with the individual nodes.

The classification algorithm is then modified to take into account these weights. Namely, for the accumulated relevance of the topics, it multiplies the number of occurrences of a word (or subtopic) by the weight w_k^j of the link between the word and the topic, and then multiplies the result by the weight w^j of the topic itself.

4 Relevance Weights

The first type of weights is associated with the links between words and topics and between the nodes in the tree (actually, the former type is a kind of the latter since the individual words can be considered as terminal tree nodes related to the corresponding topic). For example, if the document mentions the word *carburetor*, is it about *cars*? And the word *wheel*? Intuitively, the contribution of the word *carburetor* into the topic *cars* is more than that of the word *wheel*; thus, the link between *wheel* and *cars* is assigned a less weight. The algorithm of classification takes into account these weights when compiling the accumulated relevance of the topics.

It can be shown that the weight w_k^j of such a link (between a word k and a topic j or between a topic k and its parent topic j in the tree) can defined as the mean relevance of the documents containing this word for the given topic:

$$w_k^j = \frac{\sum\limits_{i \in D} r_i^j n_i^k}{\sum\limits_{i \in D} n_i^k} \tag{1}$$

by all the available documents D, where r_i^j is the measure of relevance of the document i to the topic j, and n_i^k is the number of occurrences of the word or topic k in the document i.

Unfortunately, we are not aware of any reliable algorithm of automatic detection of the measure of the relevance of r_i^j in an independent way. Thus, such a measure is estimated manually by the expert, and then the system is trained on the set of documents.

As a practical alternative, it is often possible to estimate the weights w_k^j intuitively at the stage of preparation of the dictionary. The choice of the weight is based on the frequency of appearance of the word in "general" documents from the control corpus of the texts on "any" topic; in our case such texts were the newspaper issues.

As another practical approximation, for narrow enough themes we can take the hypothesis that the texts on this topic never occur in the control corpus (newspaper mixture). Then, given the fact that we have included the word in

the dictionary and thus there is at least one document relevant for the given topic, we can simplify the expression for the weights as follows:

$$w_k^j = \frac{1}{\sum\limits_{i \in D} n_i^k} \tag{2}$$

since the numerator of the quotient in (1) in case of narrow topics can be considered to be 1. Not surprisingly, this gives the weight of the word "voting" for a specific topic to be the less the more its frequency; for example, the articles *a* and *the* have a (nearly) zero weight for any topic, while the word *carburetor* has a high weight in any topic in which it is included.

Sometimes a rare enough word, say, a noun *bill*, in its different senses is related to different topics (*money, law, birds, geography, tools*). For a more accurate analysis, some kind of competition between senses of the word for a specific occurrence in the document is introduced. For this, the senses of the word are marked in the topic dictionaries (as $bill_1$, $bill_2$, etc.), and the weights of occurrences of such a word are to be normalized by its different senses (though the occurrences of the same sense are independent in different topics), with the weigh of an individual sense in each document being proportional to the relevance of the document for the given topic:

$$\begin{gathered} w_k \sim \sum_j r_i^j w_k^j \\ \sum_k w_k = 1 \end{gathered} \tag{3}$$

where w_k is the weight of the k-th sense of the given occurrence of the word in the given document i, w_k^j is the weight of the link between this sense of the word and the topic j, the summation in the first equation is made by all the topics, and in the second by the senses of the given word. Since r_i^j in its turn depends on w_k, to avoid iterative procedure, in practice we calculate r_i^j based on equal weights w_k.

However, the latter technique is not very important for most cases, since usually it does not change the order of the topics for a document, but only makes the difference between different topics more significant.

5 Discrimination Weights

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?". Really, with such an approach taken literally, the answer will be "all the documents 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 situation. For a biologist, the answer *crustaceans* would be the best, and for an average newspaper reader the answer *nature*.

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, i.e., that the reader is a specialist in the contents of the current database. Thus, the topic relevance weights in our system depend on the database.

The main requirement to these weights is their *discrimination power*: a topic should correspond to a (considerable) *subset* of documents, while the topics that correspond to nearly all the documents in the data base are probably useless. Thus, 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:

$$w^j = \sum_{i \in D} \left(r_i^j - M \right)^2$$
$$M = \frac{\sum_{i \in D} r_i^j}{|D|} \tag{4}$$

here M is the average value of r_i^j over the current database D, and r_i^j is determined by the former algorithm, without taking into account the value of w^j.

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

6 Applications

With the approach described above, we have implemented in the system *Classifier* several useful functions.

The system can determine what are the principle topics of the document. This corresponds to the task of classification. Also the system allows viewing the documents by topics, answering the question: for a selected topic, what documents are the most relevant? This roughly corresponds to the task of information retrieval.

An interesting application of the method is classification of the documents by similarity with respect to a given topic. Clearly, a document mentioning the use of animals for military purposes and the document mentioning feeding of animals are similar (both mention *animals*) from the point of view of a biologist, but not from the point of view of a military man they are very different. The comparison is made on the basis of the weights of the topics for the two documents.

7 Discussion and Future Work

Generally, the results obtained in our experiments show very good accordance with the classification made by human experts. However, we encountered some problems with using our method. Most of them are related with ambiguity.

Sometimes, a frequent keyword (taken out of context) proves to be important for a specific topic: the noun *well* is an important term in *petroleum extraction*, the noun *do* is a term in *hair styles*, the noun *in* in *politics*, etc. However, the expression (1) assigns too little weight to such keywords. To solve this problem, we plan to add a part of speech tagger to our system. For a more detailed analysis, we might have to add our syntactic parser to the program; however, this would greatly slow down the system.

Obviously, this does not solve all the problems of ambiguity. As we have discussed, for the words like *bill* a sophisticated and not always reliable algorithm is used; we plan to resolve the ambiguity of this type with more intelligent methods described in [4].

Though there are some problems with the accuracy of the algorithm, the results of experiments show good accordance with the opinion of human experts. The method is practical in the sense of insensibility to inaccuracies in the dictionary and in the sense of using a dictionary with very simple structure, easily trainable on manually classified collections.

References

1. Anderson, J. D., Rowley, F. A.: Building End-user Thesauri from Full Text. In: Kwasnik, B. H., Fidel, R. (eds.): Advances in Classification Research. Proceedings of the 2nd ASIS SIG/CR Classification Research Workshop, Vol. 2. Learned Information, Medford, NJ. (1992) 1-13
2. Cohen, W. W.: Learning Trees and Rules with Set-valued Features. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence (1996)
3. Cohen, W., Singer, Y.: Context-sensitive Learning Methods for Text Categorization. In: SIGIR'96 (1996)
4. Gelbukh, A.: Using a Semantic Network for Lexical and Syntactic Disambiguation. In: Proceedings of Simposium Internacional de Computación: Nuevas Aplicaciones e Innovaciones Tecnológicas en Computación. Mexico (1997) 352-366
5. Guzmán-Arenas, A.: Finding the Main Themes in a Spanish Document. Journal Expert Systems with Applications **14 (1, 2)** (1998) 139-148
6. Guzmán-Arenas, A.: Hallando los Temas Principales en un Artículo en Español. Soluciones Avanzadas **5 (45)** (1997) 58, **5 (49)** (1997) 66
7. Jacob, E. K.: Cognition and Classification: A Crossdisciplinary Approach to a Philosophy of Classification. (Abstract.) In: Maxian, B. (ed.): ASIS '94: Proceedings of the 57th ASIS Annual Meeting. Medford, NJ: Learned Information (1994) 82
8. Krowetz, B.: Homonymy and Polysemy in Information Retrieval. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (1997) 72-79
9. Lewis, D. D., Ringuette, M.: A Comparison of Two Learning Algorithms for Text Categorization. In: Third Annual Symposium on Document Analysis and Information Retrieval (1994) 81-93
10. Riloff, E., Shepherd, J.: A Corpus Based Approach for Building Semantic Lexicons. In: Proceedings of the Second Conference on Empirical Methods in Natural Language Processing (EMNLP-2) (1997)

This article was processed using the LaTeX macro package with LLNCS style