Discovering Knowledge in a Large Organization through Support Vector Machines

J. A. Gutiérrez de Mesa, L. Bengochea Martínez

Dept of Computer Science
The University of Alcalá
28801 Alcalá de Henares, Madrid, Spain
{jagutierrez,luis.bengochea}@uah.es

Abstract. Much of the information used by an organization is collected in the form of manuals, regulations, news etc. These are grouped into controlled documentary collections, which are normally digitized and accessible via a content management system. However, obtaining new knowledge from collected documents in an organization requires not only sound search and retrieval of information tools, but also the techniques to establish relationships, discover patterns and provide overall descriptions of the entire contents of the collection. This article explores the nature of knowledge and the role that occupy the documentary collections as a source of obtaining him knowledge. It also describes the collection of documents will be used along the exposure of this study and the techniques of processing information in order to obtain the desired results. This paper describes the use of computational methods, support vector machines in particular, in a large organisation for document classification.

Key words: Stemming, Indexation, Support Vector Machines, Documentation and Knowledge management.

1 Introduction

Knowledge Management (KM) as discipline has acquired importance in recent years. The number of scientific articles devoted to this discipline has increased in recent years. One of the main features of knowledge management is its heavy reliance on related disciplines such as information retrieval, data mining, databases and content management systems (CMS). It can even become the standard technology for the implementation of programs of knowledge management [1].

2 Document Collection

In this work, we used a collection of articles published by a Spanish newspaper between July 1, 2004 and June 30, 2006 (two years). The collection consisted of a total of 2,067 documents. The sum of all the words in all documents of the collection

was 883,425. The number different character was 104 and the total number of characters used in all documents was 5,441,472. Most of the documents have a length between 350 and 500 words (83%).

2.1 Modelling Documents

Regardless of who is elected one level or another in the choice of terms, in all cases are going to get vectors with many dimensions. In our case, the dimensionality of space vector is given by the number of different words that are used in each and every one of the documents that are part of the collection. Of the 883,425 words contained in the collection, 37,402 are different, which is the vocabulary *V* of the collection of 37,402 words.

$$W = \sum_{i=1}^{n} w_j = 883{,}425; \quad (1)$$

One way to reduce the dimensionality is to delete words that do not add any meaning to the text (empty words) and another way is to group words that have the same root in a single lexical (Stemming) such that the total number of different words is reduced. These two processes are described below.

2.2 Normalization Process

In order to be able to run the algorithms, the first step is to transform the documents into plain text and extract the vectors that represent each of the documents. Then, there is a standardization step to facilitate the extraction of measures, such as frequencies and being the normalization the most common operation [2].

2.3 Selection of the Vocabulary

Since the documents will be classified according to their textual content, it is possible to discard all those terms that do not provide relevant information for this purpose. Human languages include many words that are only used to articulate phrases, but do not add any meaning to the text. Also, those words have with very high frequencies. Other words less frequent but more useful for the text to discriminate on the basis of their content.

The set of words that can be regarded as irrelevant or empty for a given language, in our case Spanish, is a priori comprised of the following categories: common adjectives, articles, adverbs, prepositions, conjunctions. Interjections, pronouns, auxiliary verbs (e.g., be, can, do, etc.) and modal verbs (e.g., power, hold, sing, etc.).

Once the list of empty words is built and after their removal, we have the following values: total number of empty words is 503,198; total number no empty words: 380,227 and number of different not empty words: 36,352.

While the elimination of empty words considerably reduces the size of the text, another technique is to remove those words does not reach a certain threshold to

reduce the dimensionality of space vector [3]. This technique is based on the idea that when a term appears very infrequently in a collection of documents, their discrimination capabilities is virtually zero, so it can be ruled out at the time of building the model to represent the documents [4]. With this threshold, and after the stemming the process described in the next section, the size of vocabulary will increase from V=13,256 to be V = 6,768, i.e. get a reduction of the dimensionality of nearly only 50% by removing words that appear only once, twice or three times in the set of all documents that form the collection.

3 Stemming

The basis of a lemmatizer consists of a finite state machine that tries to represent changes in a certain suffix stem. Each suffix involves a series of rules that express how a suffix has been incorporated into the stemming. Since, there can be many variations and exceptions for the same suffix, the PLC can sometimes be quite complex. From these bases, are developing various algorithms stemming for years, such as those based on the probability that a word belongs to the class defined for a stem [5].

Almost all lemmatizers are built upon the foundation of the work by Lovin [6] in 1968 and variants such as those described by Dawson [7], Porter [8] and Chris D. Dave [9]. We have also built a Lemmatizer to apply to documents from the collection object of our study, based on the works of Porter and other more specific to the Spanish language [10].

3.1 Vector Construction

With a very large number of elements that are zero, the following the nomenclature is used to represent every element of the non-zero vector: *{Position: Value}*, where *Position* is an ordinal representing the position it occupies in the lexeme vocabulary, and *Value* is the measure of the contribution that lexeme in the full meaning of the document, D_i. Therefore, a document is represented by the vector:

$$D_i = \{w_1 : f_1, w_2 : f_2, ... \quad w_{ni} : f_{ni},\} \tag{2}$$

The metrics to be used is *TF x IDF* and vectors will be standardized ($|D_i| = 1$) so that the values of f_j will be given as:

$$f_j = \frac{TF(w_j, D_i)\log(\frac{|D_i|}{DF(w_j)})}{\sqrt{\sum_j \left[TF(w_j, D_i)\log(\frac{|D_i|}{DF(w_j)})\right]^2}} \tag{3}$$

Where $TF(p_j,D_i)$ represents the frequency with which appears lexeme that took the position p_i in the document D_i; $DF(p_j)$ is the frequency of that same lexeme in the entire collection. Applying the formula to documents, get vectors as shown in Fig. 1 and who will be that we use from this point forward.

4 Classification of documents

In our study, we use the thesaurus Eurovoc [11] for selecting the categories to which documents may be assigned. This choice was justified by the need to have a package that covers all possible areas addressed in the documents. Moreover, it has been developed by experts following strict criteria.

Prior to the construction and implementation of our own classifier based on the technique known as Support Vector Machine (SVM) [12], which is the one that obtained better results classifying documents [13] , we will give a brief description of some of the most commonly used methods for grading.

In all cases it is building a model by automatic learning from a set of documents previously tagged by an expert. The model thus constructed will be able to deduct the class to which should be given every new document unknown to be present. This type is called supervised learning, because it gives the system the list of categories to which they belong all documents of a collection. A system of unsupervised learning, which builds a model able to infer the existence of clusters of documents, and hence "discover" a class structure that is not known in advance. The action taken by this type of system will call the "grouping of documents", and will be treated in the following point, to differentiate it from the classification of documents "or" text categorization "study in this point.

4.1 Support Vector Machines

Recent studies [14], [15], [16], [17] show that Support Vector Machines (SMV) are the preferred method for text classification. Unlike other methods, SVM can work efficiently with thousand of dimensions whereas in other classifiers, when there is a large number of attributes with little discrimination power, attributes need to be discarded by some preprocessing filters affecting their performance [18]. However, despite their high accuracy documented in numerous publications, and perhaps because of their complexity, SVMs have failed to completely replace simpler methods of automatic classification such as Naïve-Bayes [19].

SVM is based on the concept of minimizing risk structural which is found in the vector space which is represented as vectors documents, hyperplane separating those who belong to two different categories, and also do so with the greatest possible margin of separation. The position in space, occupy any new document, the class to determine who should be allocated. It is therefore a classifier binary and to build a multi classifier must be calculated so as hyperplans classes there.

To carry out the classification of the collection, we are going to use the program package SVM light [20] that allow us to employ algorithms SVM learning, with different parameters and kernel functions, to suit the nature of our problem. However,

how to use through orders or commands in text mode has lifted us to develop a GUI to implement the programs. In the preparation phase are formed vectors that are going to represent the set of documents or evidence of learning, properly labelled according to their membership in the class for which we construct the grid. Through panel "svm_learn" (Fig. 1) allows the user to execute the learning module with the options you choose. To carry out this operation is necessary and at least one file of learning. The outcome of this panel will be a file with the model for classifying built.

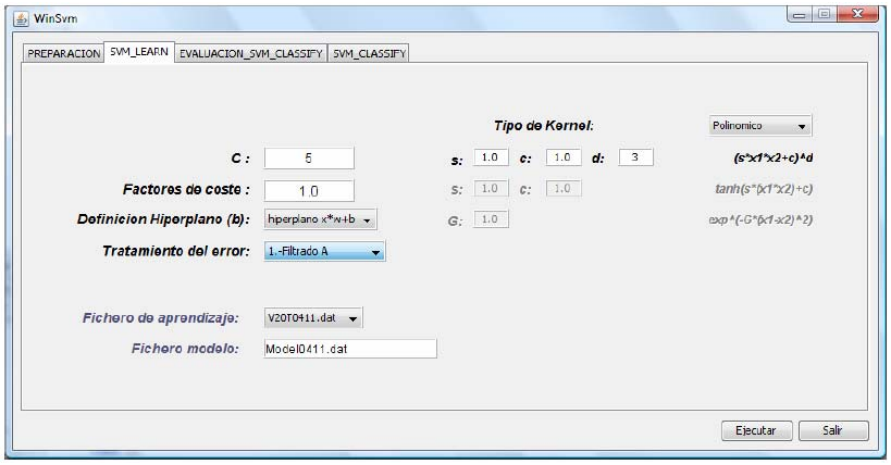

Fig. 1. Panel WinSvm for obtaining model in the training phase.

5 Experimental Results

To calculate the optimal values to apply when constructing models for qualifying, we chose the three categories that belong to a larger number of documents on joint training. These are: "04 Life policy" "08 International relations" and "28 Social Affairs". In order to have two sets of classified documents manually, one for the training phase and the other to check the behaviour of the model built, the initial set of 104 documents have been divided into two sets of 52 papers each. It has been tested with the four types of kernel function possible and the results are shown in Table 1.

Table 1. Classification with 52 papers training.

Model construction				**Test over 52**			
Category	Kernel	Iterant.	Kern Evl.	%Success	#Failures	Precision	Recall
04	Lineal	17	1556	69,23	16	72,22	54,17
04	Polynomic	19	3044	71,15	15	73,68	58,33
04	Sigmoid	9	2490	46,15	28	46,15	100,00
04	RBF	20	3099	73,08	14	75,00	62,50
08	Lineal	19	1666	71,15	15	60,00	35,29
08	Polynomic	23	3264	71,15	15	75,00	17,65
08	Sigmoid	16	2870	67,31	17	-	-
08	RBF	26	3429	67,31	17	50,00	5,88
28	Lineal	19	1666	71,15	15	-	-
28	Polynomic	26	3429	73,08	14	100,00	6,67
28	Sigmoid	17	2934	71,15	15	-	-
28	RBF	22	3209	71,15	15	-	-

The kernel function used to construct the model of learning, in all cases is SVM^{light}, a very efficient algorithm in relation to CPU time. Therefore, the only criterion we should look to choose between the modes is the rate of accuracy, leaving aside other considerations such as the number of iterations or the number of reviews of the kernel function used. Table 1 shows that is the best-performing kernel functions are linear functions and polynomic, with a slightly advantage for the latter. Therefore, for the construction of the binder will use a polynomic kernel $(\vec{x}_1 * \vec{x}_2 + 1)^3$.

5 The Emergence the New Knowledge

There are several algorithms to identify relevant phrases in a document. The most interesting are supervised learning algorithms, as C4.5, KEA or GenEx attempting to document as a set of phrases that should be classified as relevant or irrelevant. To do so, it must provide before a set of documents belonging to a body similar to those discussed and whose relevant phrases are known in advance. From this set and through a process of training builds a table of discretization of the characteristics associated with the terms deemed relevant documents joint training.

Once the system is trained, the process of automatic identification of the terms will be considered the metadata value of a new document. It consists of the following: after a period of normalization of the text obtained a first relationship sintagmes candidates, discarding those that do not meet a number of conditions (that is not its length between a maximum and a minimum preset, which begin or end with empty words, which do not reach a minimum frequency of occurrence, etc.). . It was also put to candidates for a phase Stemming with the aim of considering only the roots of words and thus increase the value of their frequencies. Then, a discreet rate of each term, based on the following values:

- Relative frequency of occurrence of S phrase in the text in relation to the overall control (TF × IDF), as measured:

$$TF \times IDF = \frac{frecuency(S, D)}{size(D)} \times (-\log_2 \frac{1 + df(S)}{N}) \quad (\mathbf{4})$$

  where frequency *(S, D)* is the number of times the phrase appeared in the paper *S D* size *(D)* is the number of words that has the document, df *(S)* indicates the number of documents corpus overall contain the term *S* (adds 1 to avoid log 0) and *N* is the total number of documents in the overall corpus).
- Distance, in the words from the beginning of the text until the first appearance of the words. The result is a number between 0 and 1 which represents the portion of the document that precedes the first appearance of a word:

$$distance = \frac{first\ time\ to\ see\ a\ sintagme}{total\ words\ of\ the\ document} \quad (\mathbf{5})$$

- Frequency with which he has already been considered as relevant among the objects of all control. This measure is known as frequency *k* and the underlying idea is that a word candidate is more likely to be relevant if it has been found as other relevant documents corpus training.

For the construction program, we started KEA system, developed in the "Digital Libraries and Machine Learning Labs" [21] at the University of Waikato and is distributed under the GNU Public License.

5.1 Clustering

The grouping a collection of papers has historically been perceived by the researchers as a discovery tool and to help reduce redundancy and demand cognitive [22]. A system of grouping should have the ability to assign each new document to the group most appropriate and should therefore be able to solve three problems: how to create groups, how to identify the relationships between the groups and how to keep the group system.

The most interesting approach in the form of documents and added that in addition, has the advantage of providing direct labels of the groups, is to extract the most relevant phrases for each document in the collection and use as a criterion for grouping. The relevant phrases are good descriptors of the topics covered in a document and therefore help build subspaces small, but representative of space full of documents.

The method used to form aggregates is to sort the relevant phrases by the number of documents that share, from highest to lowest. The first group of documents on this list, will form the nucleus of the first added, and the term will be shared by the label of this aggregate. Then he goes through the list of documents added and are appended documents with which it shares other relevant phrases. When this process is completed recursive, passed to the next term of the ordered list, and so complete.

During the process of forming aggregates in each category, you get the average length of the documents that form, expressed as the average number of days between

July 1, 2004 and the date of publication of document (column "Days (*Dias*)" on the Table 2).

Table 2. Aggregates from syntagms of long> = 1 in category 28

Social Issues	
Descriptor Eurovoc	***Aggregates***
2811.- Movimientos migratorios	Ley de Extranjería
2816.- Demografía y población	-
2821.- Marco social	alto el fuego política antiterrorista
2826.- Vida social	matrimonio homosexual Juan Pablo II
2831.- Cultura y religión	EE UU Benedicto XVI Bin Laden Conferencia Episcopal
2836.- Protección social	Ceuta y Melilla accidentes de tráfico
2841.- Sanidad	Severo Ochoa
2846.- Urbanismo y construcción	plan de choque

6 Conclusions

In this work, we presented the development of a bespoke computational science application that is going to be used by a large organization to classify documents. To do so, on the one hand, we presented the latest developments in the techniques of automatic classification and clustering textual documents. On the other hand, we showed how to build and validated models using a medium size collection of documents text in Spanish to perform measurements and results that were not previously available. Results show that it is possible to generate patterns of searching documents from the collection, exclusively using automatic learning techniques based on statistical methods, without having to implement other techniques of natural language processing.

References

1. Pérez-Montoro, M.: Sistemas de gestión de contenidos en la gestión del conocimiento. Textos universitaris de biblioteconomia i documentació. número 14. Facultat de Biblioteconomia i Documentació. Universitat de Barcelona (2005).
2. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press. Addison-Wesley. New York (1999).
3. Yang, Y., Pedersen, J.: Intelligent information retrieval .Intelligent Systems and Their Applications, IEEE vol 14, 4 (1999).

4. Luhn, H.P.: The automatic creation of literature abstracts, IBM Journal of Research and Development, 2, 159-165 (1958).
5. Allan, J., Kumaran, G.: Details on Stemming in the Language Modeling Framework. Center for Intelligent Information Retrieval. Department of Computer Science. University of Massachusetts Amherst. Technical Report No. IR289 (2001).
6. Lovins, J. B.: Development of a Stemming Algorithm. Mechanical translation and computational linguistics, 11, pp 22-31 (1968).
7. Dawson, J.: Suffix removal and word conflation. Bulletin of the Association for Literary & Linguistic Computing, pp. 33-46 (1974).
8. Porter, M.F.: An algorithm for suffix stripping. Originally published in Program, 14 no. 3, pp 130-137 (1980).
9. Paice, Chris D. "Another stemmer". ACM SIGIR Forum archive, v. 24, pp. 56-61 (1980).
10. Figuerola, C.G.; Gómez, R., López, E.: Stemming and n-grams in Spanish: An evaluation of their impact on IR. Journal of Information Science vol. 26, pp. 461-467.
11. Eurovoc: Tesauro Eurovoc. Presentación alfabética permutada. Edición 4.2 - Lengua española. ISSN 1725-426. Comunidades Europeas, (2006).
12. Joachims,T.: Leaning to classify text using SVM Methods Theory and Algorithms. Kluwer Academic Publishers (2001).
13. Sebastiani, F. Classification of text, automatic: Keith Brown (ed.), The Encyclopedia of Language and Linguistics, vol. 14, Elsevier Science Publishers, Amsterdam (2006).
14. Flach,Peter A.: On the state of the art in machine learning: A personal review. Artificial Intelligence, 131, pp. 199-222 (2001).
15. Yang, Y., Zhang, J., Kisiel, B.: A scalability analysis of classifiers in text categorization. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval. ACM Press (2003).
16. Lewis, David D., Yang, Yiming, Rose, Tony G., Li, Fan.: RCV1: A New Benchmark Collection for Text Categorization Research". The Journal of Machine Learning Research, vol. 5. MIT Press (2004).
17. Lai, Chin-Chin.: An empirical study of three machine learning methods for spam filtering. Knowledge-Based Systems, vol 20, pp. 249-254 (2007).
18. Gabrilovich, E., Markovitch, S.: Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4.5. Proceedings of the twenty-first international conference on Machine learning ICML '04. ACM Press (2004).
19. Gayo Avello, D.: BlindLight - Una nueva técnica para procesamiento de texto no estructurado mediante vectores de n-gramas de longitud variable con aplicación a diversas tareas de tratamiento de lenguaje natural". Univ. Oviedo. Dpto. de Informática (2005).
20. Joachims, T.: Leaning to classify text using SVM Methods Theory and Algorithms. Kluwer Academic Publishers (2001).
21 Frenk *et al*.: Domain-specific keyphrase extraction. Proc. Sixteenth Int. Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Francisco, CA, pp. 668-673. (1999)
22. Roussinov, D. and Chen, H.: "A Scalable Self-organizing Map Algorithm for Textual Classification: Neural Network Approach to Thesaurus Generation". Communication and Cognition volume 15, number 1-2, pp. 81-112 (1998).