Clustering and retrieval of Spanish news documents using self organizing maps[*]

Javier Fernández, Ricardo Mones, Irene Díaz, José Ranilla, Elías F. Combarro
Artificial Intelligence Center
University of Oviedo (Spain)
ir@aic.uniovi.es

Abstract

In this, our first participation in the Cross Language Evalution Forum, we want to test the performance in Information Retrieval of a clustering method which uses Self Organizing Maps.

1 Introduction

Many approaches to Information Retrieval focus at the word level. Thus, indices of words are built, synonyms of words are used to expand queries and so on.

In this work for the Cross Language Evaluation Forum we want to test an approach which focuses more on the document level. First, we create clusters of similar documents and then we search for the cluster more similar to a given query. This method should have several advantages:

1. The search is faster, since the number of elements to compare with the query is decreased

2. Similar documents are clustered together and they can be retrieved even if they contain no words of the query (without the use of a thesaurus)

3. In addition to an Information Retrieval system a hierarchy of the documents is built

In this paper we describe a very simple system of that kind, based on Self-Organizing Maps [3]. These maps have been used successfully for a series of tasks in Information Retrieval [7, 4, 5, 6].

We test the performance of the system for the Monolingual Spanish task of the 2003 edition of the Cross Language Evaluation Forum, which consists of searching for 60 different topics over the news published during 1994 and 1995 by the EFE agency.

The rest of the paper is organized as follows. In section 2, we briefly describe similar approaches that can be found in the literature. Section 3 is devoted to describing the system. The experiments conducted to test the system are described in Section 4 and their results are presented in section 5. Finally, in section 6, we draw some conclusions and propose some ideas for further research.

2 Previous works

Self-Organizing Maps ([3]) or simply SOMs, are methods for unsupervised clustering of elements. These elements are represented by numerical vectors which are used, in a training phase, to adjust the vectors asociated to a set of neurons. The neurons are usually situated in a 2-dimensional topological map- This topology has influence in the training phase, since the values of a neuron are modified together with those of its neighbours.

[*] The research reported in this paper has been supported in part under FICYT grant PR-01-GE-15.

These maps have been succesfully use in a wide variety of tasks [2, 3], including document management [7, 4, 5] and Information Retrieval [6].

The usual application of SOMs is to cluster similar documents together. Then, the search is performed over the neurons instead of over the documents themselves and then the computational complexity is reduced several orders of magnitude.

This clustering has also the advatange of disambiguating the meaning of words (since the context of the word is considered altogether) and that has proven to provide an improvement in the retrieval of relevant documents [6] at least compared with simple classical methods as Salton's vector space model [10] and Latent Semantic Indexing [1].

3 The system

The number of documents that we are going to use in the experiments (see section 4), is over 300 hundred times bigger than others used in the referred literature with the same system [6]. This makes infeasible the use of all the words of the corpus to represent the documents, and makes necessary the selection of the most representative words or features.

Then, the main difference of our system with previous approaches (see section 2) is the use of two different representations of the documents. Since we are dealing with documents which come from a news agency, we consider that proper nouns are specially important in this framework.

The frequencies and distribution over the documents of proper nouns and other words are quite different. If we use just one vocabulary, then most proper nouns will be dropped when we filter the vocabulary, resulting in a serious limitation in the retrieval phase. For this reason, we will deal with two separate vocabularies: one of proper nouns and another one of common words.

Each document is represented as two vectors (see [9]). The first one has as components the number of times that each member of the vocabulary of proper nouns appear in the document; the second one is similar, but with respect to the vocabulary of common words.

Then, two different groups of clusters are constructed using the two different groups of vectors representing the documents. When we are given a query, it is confronted to both groups of clusters and the results are combined to give the final set of documents considered relevant for the query.

The whole process involves the following steps:

1. Extraction of the vocabularies
2. Selection of the features representing the documents
3. Clustering of the documents
4. Comparison of the query with the clusters
5. Combination of the results

They are described in more detail in the following sections.

3.1 Extraction of the vocabularies

We iterate through all the documents in the collection to get the two vocabularies: the one of proper nouns and the one of common words.

The first step consists of deciding whether a word is a proper noun or not. To this extent, some complex algorithms have been developed (the Named Entity Task of the Message Understanding Conferences was devoted to this problem) but we take advantage of the capitalization in Spanish language to use a simple method: a word which begins with capital letter and is not a start of a sentence is a proper noun; a word all in lower case letters is a common word; any other word is declared ambiguous and its type is decided after all other words have been processed. If the ambiguous word appears somewhere else in the document as proper noun then it is a proper noun. If not, it is considered a common word. This process is depicted in Figure 1.

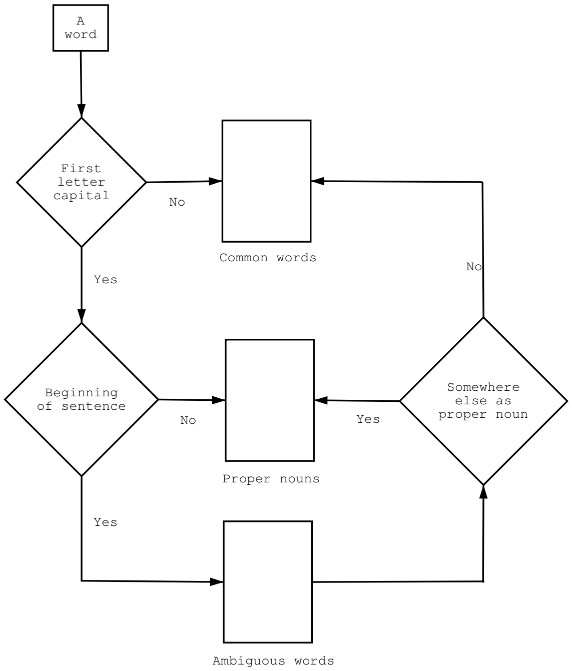

Figure 1: Deciding whether a word is a proper noun

During this process, all words are normalized. They all are put in lower case and diacritical marks are eliminated[1]. Also, common words pass through a process of stemming, performed with a stemmer for the Spanish language based on that of Porter [8].

From the vocabularies, we eliminate all the stop-words. We consider two different lists of stop-words, one for proper nouns and another one for common words. The one for proper nouns is much shorter than the other, since many words that have empty meaning when used as proper nouns. For instance, "más", is the Spanish for more, and is a stop-word when used as common word, but it is also a surname, and thus should not be eliminated when used as proper noun.

The whole process is summarized in Figure 2.

3.2 Selection of the features representing the documents

In most cases the number of words in the vocabularies is very big. This makes almost impossible the use of some algorithms because the dimensionality of the vectors that represent the documents is too high. Also, most of the words are really non-informative.

For these reasons, we select only a small number of the words in the vocabularies to actually represent the documents in which they appear. To choose the relevant words, we use two well-known filtering measures: *tfxidf* (see [9]) for the vocabulary of common words and *df* (the number of documents in which the word appears) for proper nouns.

We use two different measures for the vocabularies since their behavior is not the same. While a proper noun which appears in many documents is clearly important, a common word which is a lot of different documents will likely be non-informative. That is why document frequency is considered important to select proper nouns but not for other words.

[1] As they are used by the stemmer, diacritical mark elimination for common words is delayed up to the stemming phase has been finished

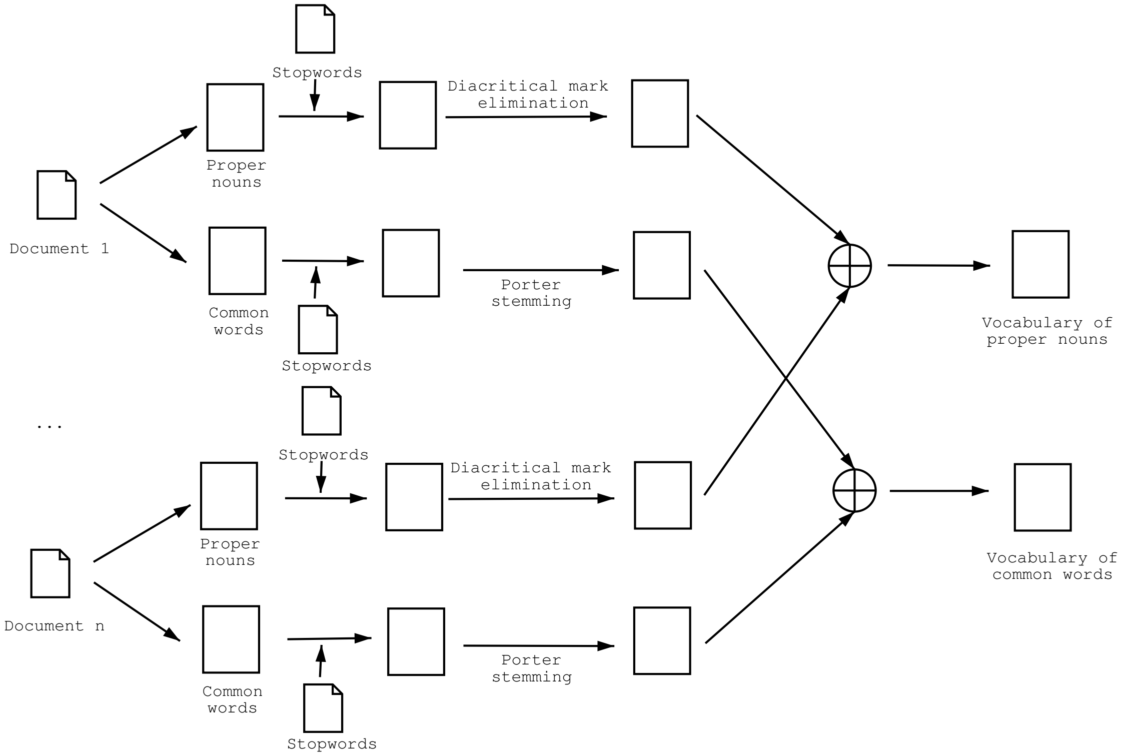

Figure 2: Extraction of the vocabularies

3.3 Clustering of the documents

Since we have two different kinds of representations of the documents, we train two different networks, one with each representation. For the training we do not use all the documents, but a number of them which contain enough appearances of the words in the corresponding vocabulary. If the word of the vocabulary which appears in less documents has $df = x$ then we impose that the number of documents selected for the training must be big enough for all the words of the vocabulary to appear in at least $\frac{1}{4}x$ training documents.

The size of the SOM can be varied in order to obtain bigger or smaller clusters.

3.4 Comparison of the query with the clusters

Given a query we represent it in the same way that we represent the documents. Thus, we will have two different vectors for each query. Each of them is compared with the corresponding network, finding the neuron which is closest to the vector according to the cosine distance defined by

$$d(q, v) = \frac{\sum_{i=1}^{n} q_i v_i}{\sqrt{\sum_{i=1}^{n} q_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}$$

where q_i are the components of the vector representation of the query, v_i are the components of the vector associated to the neuron and n is the size of the vectors.

All the documents mapped to the neuron are regarded as relevant to the query. If more documents are needed, the process is repeated with the next closest neuron.

3.5 Combination of the results

After that, we will have a number of documents which have been found relevant to the query with the network of proper nouns and another set of documents which come from the network of common words. To combine the results obtained, we average the distances of the two representations of the documents to the corresponding vectors representing the query.

4 The experiments

We have used the system described in section 3 for the Monolingual Spanish task, which consists of 60 queries over all the 454042 documents published by the news agency EFE during the years 1994 and 1995.

After processing the documents, we have obtained 277196 proper nouns and 144652 common words. Of these, we have retained all the proper nouns that appear in at least 100 different documents and the 5% of common words with the highest $tfxidf$. This makes a total of 9924 proper nouns and 7233 common words.

We have trained SOMs of three different sizes: 25x25, 50x50 and 75x75 neurons. Thus, we can test the influence of the number of documents in the clusters. This number was in average 726.467 for the 25x25 networks, 181.617 for 50x50 and 80.718 for 75x75.

5 The results

5.1 Official runs

After submitting the results of the official experiments, we detected a serious bug in the implementation of the algorithm that caused that the output of the system was completely wrong. In fact, the experiments carried out, with the networks of sizes 25x25 and 50x50, returned 60000 documents each, of which only four were relevant for any query.

5.2 Unofficial runs

Once the bug was detected and corrected, we repeated the official experiments and also conducted some other to perform a more thorough test of the system.

Since it seems that the number of documents taken from each network is a very important parameter (see [6]), we have tested its influence in first place. For that reason, we have fixed the set of the networks to 50x50 (neither too small, not too big) and performed experiments retrieving a number of documents from each network ranging from 2000 up to 30000. Then, the distances of all these documents to query are computed and only the 1000 closest ones are retained. In Table 1 we summarize the results of these runs.

We can see that there is an improvement if we take more documents from the networks (though from 7000 to 10000 there is a small decrease), up to 15000 documents. Also, from 20000 documents the overall performance decreases again. Then, we have decided to use 15000 documents from each of the networks in the rest of the experiments.

With this number of documents, we have perfomed experiments with the different sizes of networks. Results can be seen in Table 2.

In view of the results, one can argue that the performance could increase if the size of the network is set to 100x100 or even bigger (up to some point). Unfortunately, nowadays we lack the computer power needed to test the method with those settings. The amount of memory needed to load all the train documents and to store the neurons of the network is too big. A possible solution is to split the train documents into smaller subsets and train the neurons in a sort of batch process, where the output network of a training is the input of the next one. However, we have decided not to do so because it is not clear that the results obtained are really comparable

Table 1: Influence of the number of documents

Documents per network	R-precision
2000	0.0454
3000	0.0459
4000	0.0461
5000	0.0498
6000	0.0509
7000	0.0530
8000	0.0507
9000	0.0498
10000	0.0496
15000	0.0629
20000	0.0621
25000	0.0618
30000	0.0619
All	0.0365

Table 2: Influence of the size of the networks

Size of network	R-precision
$25x25$	0.0460
$50x50$	0.0629
$75x75$	0.0665

Also notice that, in every case, the use of the SOMs to preselect a number of documents gets better results than the use of the traditional model in which simply the documents closest to the query are selected (see last row of Table 1). This confirms the results obtained in [6] for a much smaller collection.

However, we consider that these results are very poor and that it is needed to consider modifications of the method in order to get satisfactory results.

6 Conclusions and future work

The experiments shown here suggest that the use of SOMs in IR can be of help (cf. [6]), even in situations when the amount of information is very big and the number of relevant documents is small.

To improve the results, we will study modifications of the method. It seems that the size of the netwoks used and the number of features selected for the representation of the documents have a crutial influence. For this reason, we plan to perform experiments with different values of this parameters (using batch mode for the biggest ones).

The way that the results of the two netwoks are combined is also important, so we consider important to investigate other methods, including weighted averages and earlier combinations of the vocabularies and of the networks.

References

[1] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic indexing. *Journal of the American Society for Information Science*, 41(6):391–

407, 1990.

[2] S. Kaski, J. Kangas, and T. Kohonen. Bibliography of self-organizing map (SOM) papers: 1981-1997. *Neural Computing Surveys*, 1(3&4):1–176, 1998.

[3] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Science.* Springer Verlag, 2001.

[4] Teuvo Kohonen, Samuel Kaski, Krista Lagus, and Timo Honkela. Very large two-level SOM for the browsing of newsgroups. In C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff, editors, *Proceedings of ICANN96, International Conference on Artificial Neural Networks, Bochum, Germany, July 16-19, 1996*, Lecture Notes in Computer Science, vol. 1112, pages 269–274. Springer, Berlin, 1996.

[5] Teuvo Kohonen, Samuel Kaski, Krista Lagus, Jarkko Salojärvi, Jukka Honkela, Vesa Paatero, and Antti Saarela. Self organization of a massive text document collection. In Erkki Oja and Samuel Kaski, editors, *Kohonen Maps*, pages 171–182. Elsevier, Amsterdam, 1999.

[6] K. Lagus. Text retrieval using self-organized document maps. *Neural Processing Letters*, 15(1):21–29, 2002.

[7] Krista Lagus, Timo Honkela, Samuel Kaski, and Teuvo Kohonen. Self-organizing maps of document collections: A new approach to interactive exploration. In Evangelios Simoudis, Jiawei Han, and Usama Fayyad, editors, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 238–243. AAAI Press, Menlo Park, California, 1996.

[8] M. F. Porter. An algorithm for suffix stripping. *Program (Automated Library and Information Systems)*, 14(3):130–137, 1980.

[9] G. Salton and M. J. McGill. *An introduction to modern information retrieval.* McGraw-Hill, 1983.

[10] G. Salton, A. Wong, and C.S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.