Learning by doing: A baseline approach to the clustering of web people search results

José C. González, Pablo Maté, Laura Vadillo , Rocío Sotomayor, Álvaro Carrera

E.T.S.I. Telecomunicación
Universidad Politécnica de Madrid
E-28040 Madrid (Spain)
+34 91 5495700 x.3030

josecarlos.gonzalez@upm.es, materio@hotmail.com, lvadillo@alumnos.upm.es, rociosotomayor@gmail.com, carrerabarroso@gmail.com

ABSTRACT

This paper has two readings. The first one consists in the description of the technical approach followed in the development of a system participating at the 2009 Web People Search Clustering Task [2]. The second reading is about an educational experiment by which a group of 12 students following a course on Intelligent Systems (in the last year of their MSc degree in Telecommunication Engineering) have fully developed such a system. Technically, the system can be defined as a baseline for future research, with basic and easy to implement approaches for every system module.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval – Information Search and Retrival – *clustering.*

K.3.2 [**Computing Milieux**]: Computers and Education – Computer and Information Science Education – *computer science education*

General Terms

Algorithms, Documentation, Experimentation.

Keywords

Search, clustering, information retrieval, named entity, named entities recognition,

1. INTRODUCTION

This work has been carried out by a group of 12 students enrolled in a course on Intelligent Systems, in the 5th and last year of their degree in Telecommunication Engineering at Universidad Politécnica de Madrid. The work was conceived as a group assignment for a course module on "Applications of Intelligent Systems", being initially granted with 1,6 credits (equivalent to 16 hours of class, plus group work). The offer to participate at the WePS clustering task, as an alternative approach to other learning methods followed in previous years was accepted by 12 of the 14 students registered for the course. For most of them, this was a "first time" experience: the first activity ever done linked to research, the first time for collaborating in such a big group, the first time participating in a professional international event… and the first time writing a technical paper. One student, Pablo Maté, was appointed coordinator, and the following teams were established: pre-processing, indexing, clustering and experimentation and delivery of results. Weekly meetings were hold for coordination purposes along six weeks. Unfortunately, there was no time for improving the system beyond parameters tuning, once first results were obtained.

The technical approach consisted in starting by carrying out a simple preprocessing filtering, leaving out html tags and paying special attention to the named entities in each file. For that purpose, a strategy was adopted to find the different ways a name could appear. The initial goal (not fully tested) was creating an index with one representative, normalized form of the entities identified in every archive.

Lucene [1][1], the information retrieval engine, was imposed as the core of the system. On the educational side of this experience, it was considered that knowing this widespread IR tool was an important side effect. The use of Lucene was going to facilitate the processing of texts in different languages [3][4], besides the chances to build different indexes and search strategies for named entities and normal text.

The approach to clustering had to be kept simple. The use of special clustering libraries or applications would have imposed too much learning time. So, a naïve approach based in simple "similarity" relations between pairs of files was conceived. It was known in advance that this approach would not work properly without considering attributes like page length (in number of words), but more sophisticated mechanisms would be tried in case time allowed it.

The last work package consisted in creating the output format for delivering results, carrying out system runs and, if possible, providing guidelines to the other work packages for system improving. This late goal was not achieved due to time pressures. However, some parameterization of the clustering system had been foreseen, providing opportunities for some level of system tuning.

2. PREPROCESSING

The first step implementing our proposal for the WePS clustering task was to adapt the input documents given by the organization to the format compatible with the tools that were going to be used. This process is divided in two stages: turning HTML files into simple text files and preprocessing the information to make easier next phases. Let's describe these two stages.

1. HTML to text. To facilitate the analysis of the text, we need to turn the HTML files given into simple text files. To do that, we have used the class "StringExtractor" from the "HTML Parser" library (available in http://htmlparser.sourceforge.net/). Given the path of an HTML file, this class obtains the textual contents of the

page, eliminating HTML tags. Then, we can write those textual contents into a text file.

2. Preprocessing files. Once we have obtained the textual contents, we proceed to process the information. In our case, we are going to obtain every possible form in which a person name can appear in a HTML file. The process consists of building an "Entity" for each person name given. For each "Entity", we define a pattern range of forms in which a name can be found. Next, we define an "Analyzer" which searches those patterns in the textual contents of the HTML page. Finally, a preprocessing file is written for each HTML file. This preprocessing file is structured as follows: the first line indicates the path of the associated HTML file; then each line contains one of the forms in which the person name appears in the HTML file.

Let's see an example. We start from a file structure based on folders associated to person names containing different HTML files in which the name appears. Lets take a specific name, John Kennedy. Each HTML file produces two text files. One contains the textual contents of the HTML file, and the other contains the path and the different forms of the name, i.e. :

{./John Kennedy/webpages/1/index.html}
{John F. Kennedy}
{John P. Kennedy}
{JFK}
{Kennedy}

These patterns will be useful when we define name search in the next stages of the task.

3. INDEXING

In this work package, the pre-processed data (the plain text obtained from HTML files) are indexed as a first step to compute similarity between pairs of pages.

To achieve information retrieval functionality and performance, Apache Lucene is used. Lucene provides some internal tools for organizing and querying indexed data. Furthermore, we use an stemming algorithm (Porter Stemming Algorithm) plus stopwords filtering to discard common or no-content words, like common or auxiliary verbs, pronouns, etc.

The description of the process of the system is the following. Firstly, we mark each document with three fields.

- Path: the location of a document in the file system, to be able to locate the document for constructing the clusters.
- Names: names of people who appear in a document (with all their variants).
- Content: content of a document (plain text).

Secondly, each plain text document is indexed into the system. But, through Porter Stemming Algorithm, some words are not considered in future searches.

Finally, the whole information is indexed and the system is ready to start the next phase: similarity computation.

This phase involves a heavy computing load and it spends most of processing time.

4. CLUSTERING

Once the files are indexed and sorted in Lucene, we have to compute its similarity degree in person names with the aim of forming clusters.

In this case, we use a technique based in similarity between documents, which is determined by the number of words in common in analyzed documents, compared by pairs.

First, all the documents including a mention to each person name under consideration are obtained. We analyze each of the documents associated with the name of the person, forming up its array of frequent words (AFW). For this purpose, the Lucene QueryTermVector class is used.

This resulting array orders the words from highest (W1) to lowest (WF) frequency. The most frequent terms and the frequency they appear in each text can be obtained through the functions getTerms () and getFrequencies ().

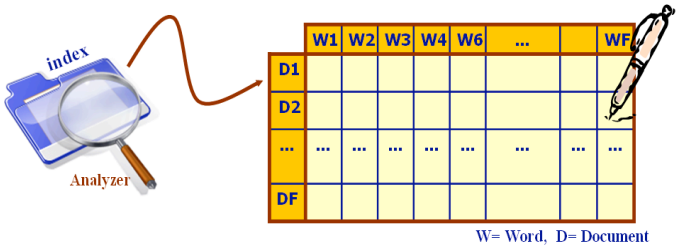

Figure 1. Two-dimensional array showing documents and their most frequent words.

The figure above shows the result of this first phase. As can be seen, the result of this phase is a two-dimensional array of documents and most frequent words.

Once the array of frequent words (AFW) from each of the documents is obtained, we compare vectors of the different files. The AFW of a document is compared with all other AFW for all other files under consideration.

The similarity degree between two AFW (corresponding to two different documents) is computed as the result of the amount of coincidences in these vectors. Each frequent word of every document is compared with all the terms of the other documents. When a particular term is included in the rows corresponding to two different documents, the score expressing the similarity between both documents is increased in one unit.

The result of this second phase is a two-dimensional array that compares the similarity of pairs of documents. Each cell in the array shows the number of frequent words shared by two individual documents. This resulting array is a symmetrical two-dimensional array, as shown in figure 2

Once the degree of similarity between two documents is obtained, it is possible to guess whether they belong to the same cluster or not. A simple way to decide if two web pages are in fact associated with the same person consists in setting a similarity threshold (ST). Above the threshold, it will be decided that both pages refer to the very same person, so being included in the same cluster.

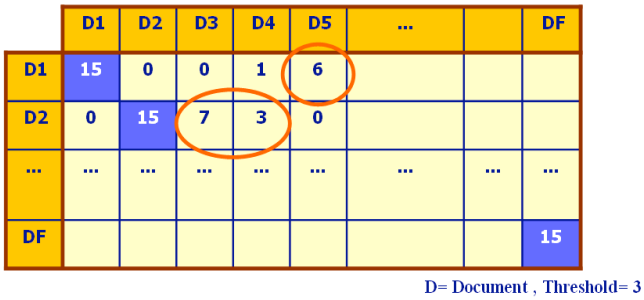

Figure 2. Document similarity Matrix

In the example case depicted in Figure 2, considering a ST of "3", the resulting clusters are as follows: D2-D3-D4, included in one of the clusters, and D1-D5, included in a second cluster.

The result of the comparison is collected in a new table of clusters. Each of the clusters should incorporate all the documents making reference to the same person. Different clusters should contain documents about different persons, even when they share the same name.

This process is repeated for each of the person names of the given set. For each of them, the two dimensional array of documents containing its identifier and the AFW has to be generated. Subsequently the similarity matrix is computed and the clusters are identified through the similarity threshold.

The results are delivered as an xml document showing the clusters linked to each person, and the documents included within each cluster.

5. EVALUATION

The evaluation stage was planned to consist in the analysis of the results obtained by the system. Unfortunately, because of a lack of time to feed-back the firsts stages with the final scores and the time needed to index all the data, the evaluation stage was reduced to the testing of different combinations of clustering parameters and pick the best results, comparing them to the zero-work procedures like ALL IN ONE or ONE IN ONE and the last year final results.

The results of the variation between number of words to consider and the decision threshold in the clustering part are shown in figure 3, which shows the threshold from 0 to 9 and the number of words considered in the other axis.

	0	1	2	3	4	5	6	7	8	9
1	0,42	0,37	0,37	0,37	0,37	0,37	0,37	0,37	0,37	0,37
11	0,57	0,51	0,46	0,42	0,41	0,4	0,39	0,39	0,39	0,38
21	0,6	0,57	0,54	0,5	0,47	0,44	0,43	0,42	0,41	0,4
31	0,61	0,59	0,57	0,55	0,52	0,5	0,47	0,45	0,43	0,42
41	0,61	0,6	0,59	0,58	0,57	0,54	0,51	0,5	0,47	0,45
51	0,61	0,61	0,6	0,59	0,59	0,57	0,55	0,54	0,52	0,49
61	0,61	0,61	0,6	0,61	0,6	0,59	0,45			

Figure 3. Precision achieved as a function of the number of most frequent words (in the range 1-61) and the similarity threshold for clustering documents together (0-9 words in common).

The results of the official runs submitted to the conference organization are related to specific decisions on the values of these parameters:

- Number of most frequent words selected for the characterization of documents.
- Number of words in common between documents to be considered in the same cluster (matching threshold)

The official results obtained for each run related to the parameter selection are presented in figure 4, showing a low sensibility of the results with respect to the values of these parameters.

Run id.	#Frequent words	Matching threshold	FMeasure 0.5_BEP-BER
UPM_SINT_1	31	1	0,53
UPM_SINT_2	41	1	0,53
UPM_SINT_3	51	2	0,52
UPM_SINT_4	**61**	**3**	**0,54**
UPM_SINT_5	110	1	0,53

Figure 4. Official results

It is easy to notice that the best results arise with a high number of words considered but a tiny threshold. As long as the number of words considered increases, the score increases too, and when the threshold increases, the score decreases. In the limit, this would led to the ONE IN ONE algorithm, showing the deficiencies of this approach.

One of the improvements that are intended to be included in subsequent experiments is the change of the vector of common words. It has been observed that, in these vectors, the first most frequent words do not provide usually significant value in identifying the person.

Another obvious improvement in the document processing refinement process and the subsequent construction of clusters is to include in the comparative specific features extracted from the text. (E.g., analyzing the dates of birth or death, if they are part of the information content of the web page). This may provide additional relevant information that could help in deciding about clustering. Of course, the extraction of semantically relevant information is not a simple improvement, especially if it involves temporal reasoning.

Another improvement would be trying to include a vocabulary of terms relating to professions, hobbies, etc., that would allow to define aspects of life in a more specific way. Thus, one could distinguish more clearly what pages belong to a specific person, depending on the topics of news, biography and other items.

6. CONCLUSIONS AND FUTURE WORKS

The baseline system built until now illustrates some obvious conclusions:

1. The extraction of features characterizing people, like contact pointers, fields of activity, dates, names of other entities (persons, organizations, places, etc.) linked within a document, are of paramount importance in the task of clustering web pages.

2. Content structure of a particular web page reveals its purpose. So, it is easy for humans to distinguish among a contact page, a page of bibliography, a piece of news, an entry of an encyclopedia, a blog entry, etc. Considering the implicit purpose of a page may be valuable also for clustering.

Regarding the educational experience carried out with a team of 12 undergraduate students with no previous knowledge of the field of Information Retrieval, the results have been encouraging. The level of involvement in the process of developing the system was high, despite the low level of accreditation initially planned for the work, as it was only an assignment for a short module of a course on Intelligent Systems. It means that the motivation was high. The opportunity to have an initial research experience in an international environment was somewhat exciting for them. Regarding extra curricular skills, the project has provided an unusual environment for real work group, besides the possibility of carrying out purposeful technical presentation and writing. Now, the challenge consists in how to benefit from the results and conclusions of this year for other teams of students in subsequent WePS campaigns.

7. ACKNOWLEDGMENTS

Our thanks to all the other students who participated in the development of this system. In alphabetical order: Giacomo Bartoloni, Isabel Bau, Miguel Coronado, Mathieu Fiolet, Juan L. García, Alvaro Martín, Andrej Remic, Jorge A. Rodríguez.

This work has been partially supported by the Spanish R+D National Plan, by means of the project BRAVO (Multilingual and Multimodal Answers Advanced Search – Information Retrieval), TIN2007-67407-C03-03 and by Madrid R+D Regional Plan, by means of the project MAVIR (Enhancing the Access and the Visibility of Networked Multilingual Information for the Community of Madrid), S-0505/TIC/000267.

8. REFERENCES

[1] Apache Lucene project. On line http://lucene.apache.org [Visited 22/02/2009].

[2] Artiles, Javier, Gonzalo, Julio and Sekine, Satoshi. WePS 2 Evaluation Campaign: overview of the Web People Search Clustering Task. In 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, April 2009.

[3] Porter, Martin. Snowball stemmers and resources page. On line http://www.snowball.tartarus.org [Visited 22/02/2009].

[4] University of Neuchatel. Page of resources for CLEF (Stopwords, transliteration, stemmers …). On line http://www.unine.ch/info/clef [Visited 22/02/2009].