THE IMPORTANCE OF THE DIFFERENCE IN TEXT TYPES TO KEYWORD EXTRACTION: EVALUATING A MECHANISM

Christos Bouras
Research Academic Computer Technology Institute and
Computer Engineering and Informatics Dept., University of Patras,
N. Kazantzaki, University Campus, GR26500, Rio, Patras
bouras@cti.gr

Charis Dimitriou
Computer Engineering and Informatics Dept., University of Patras,
Building B, University Campus,GR26500, Rio, Patras
dimitrio@ceid.upatras.gr

Vassilis Poulopoulos
Research Academic Computer Technology Institute and
Computer Engineering and Informatics Dept., University of Patras,
N. Kazantzaki, University Campus, GR26500, Rio, Patras
poulop@ceid.upatras.gr

Vassilis Tsogkas
Computer Engineering and Informatics Dept., University of Patras,
Building B, University Campus,GR26500, Rio, Patras
tsogkas@ceid.upatras.gr

Abstract - *Information exists in every aspect of our life. The expansion of the web has helped to this direction. The web feeds us with enormous information and the widespread use of computers and other hardware appliances has lead us to a state where we have a lot of information in our hands, but many times it is useless. People are not able to find information that they really need but already own. How many times have you tried to find a specific article that you have, or a specific mail that you received, or even an SMS from someone saying something specific. For this reason many information retrieval techniques have been proposed and many information extraction mechanisms have been created. In this paper we will provide the experimental evaluation of a keyword extraction mechanism and how we treat different types of text (news articles, publications, e-mails). This keyword extraction mechanism is a part of a complete system that includes information retrieval, information extraction, categorization and publication of information to a personalized portal.*

Keywords: Information Extraction, Keyword Extraction, Text Preprocessing, Web Service

1. Introduction

We live in an era that information exists in every aspect of our life. We have to admit that creation of information of every type is done easiest than ever. Apart from the information that derive from digitalization of hard copied documents or analog videos the World Wide Web is a source of massive information. A very important issue that derives from the existence of enormous information is the act of searching for specific information according to our needs. This means that we have to make any kind of information “searchable”. A whole part of the researchers world has been involved with a view to create procedures that will make our life easier when we try to find some kind of information. Information Extraction is a procedure that leads to extraction of “useful” information from a document which could be a simple plain text or even a video.

Information extraction is actually some kind of translation of a document into a human understandable language. This can be achieved by transforming the content of the document into a commonly used language (for the devices), that is interoperable and independent, and it is called XML (eXtensible Mark-up Language [3]). The main scope of information extraction is to transform any kind of information that is commonly used in our ordinary world into a structured format that could be processed easily and from which we can obtain information either that we haven't observed or the ones we want to seek for. IE mechanisms are widely used in our everyday life and especially on the World Wide Web because of its chaotic nature. IE is very useful for a variety of applications that have been introduced in recent papers: research project homepage [4], geographic web documents [5], medical abstracts [6] and government reports [7].

Information extraction is the prevalent idea when we want to achieve text mining. In addition when we combine the information extraction with the use of tools for Natural Language Processing [8] we can have beneficial results according to the fields that we will apply a mechanism. We have to admit that when we implement NLP techniques we have at least one basic restriction (specific language) but we can definitely achieve better and more precise results. Useful advanced information extraction systems can be found at [1], [2], [9], [10] and on research that was done by the authors in [14] and [15].

In order to achieve information extraction a number of procedures have to be completed. The procedures include fetching the text from its source which could be a database, a file system or the World Wide Web. The text has to be transformed in order to be readable by the text processor. The text processor extracts the appropriate information and usually a taxonomy is used for the characterization of the text and for the extraction of the semantic or not information. The procedure usually has many steps, though the one that seems to be the most critical is the keyword extraction mechanism. It is a programmer's work to produce a document in that way in order to be readable by a mechanism. The taxonomy can be used only if the corresponding keywords are extracted. What remains is the keyword extraction part which will be described in this paper. Many researches has been done on keyword extraction in [11], [12] and [13].

In this paper we will specialize on the preprocessing procedure of an IE mechanism. We believe it the most crucial part of the mechanism as it is the procedure from which the basic meaning of a document derives and more specifically the keywords of it. When we talk about document we refer to plain text. This means that the input of our mechanism before the procedure of preprocessing should be plain text. Text preprocessing has 5 simple steps. Text is made to lower, punctuation is removed, words with length smaller than a limit are also removed, more words are removed according to a list of stop words and finally a stemming procedure is done in order to extract the root of each word. In each of the aforementioned steps there are some parameters that have to be taken into consideration. For example, if all the punctuation has to be removed (ex. C.P.U.), which will be the word's length limit, which will be the list of the stop words, which percentage of the keywords can be representative of the text.

The text that exists on the web can be found in various types. The text that derives from a paper is much more different than the one that derives from an e-mail. We believe that the mechanism should be able to treat in different ways the different types of text, which means that we should change the parameters of preprocessing during the procedure of information extraction from different types of text. In our paper we will present the results of the preprocessing procedure of an IE mechanism where we describe how we believe we should parameterize the system in order to perform better with different types of text.

The rest of the paper is structured as follows. In section 2 we describe the architecture of the complete mechanism that we constructed giving more emphasis on the KE mechanism that we have developed. In section 3 we describe the input and output of our mechanism and in section 4 we analyze some issues concerning the structure of information.

2. Architecture

The system described is part of a whole system that focuses on information retrieval (from the WWW), information extraction, categorization and representation of the results to the user. Every part of the system can be an individual part and can be implemented autonomously. Though, because of the nature of the system we decided to use a universal database for all the systems in order to achieve the best communication between the parts. In addition we have universal inputs and outputs for each system in order to be able to use them without having to involve any procedures apart from the specifics for each module.

In figure 1 we can view the architecture of the mechanism together with the inputs and outputs of each module. We focus on the information extraction part of the mechanism because it is the one we want to describe.

The input of the information extraction module is either pure text or XML. The text can be read from a file or from a database with the precondition that it will be plain text.

Apart from the file, a number of parameters must be given as input to the mechanism in order to:

- specify the minimum word length (all words with length smaller than the minimum will be removed),
- specify if the numbers will be stored or removed,
- specify the stop word list that will be used,
- specify the stemming algorithm that will be used ([16] and [17]).

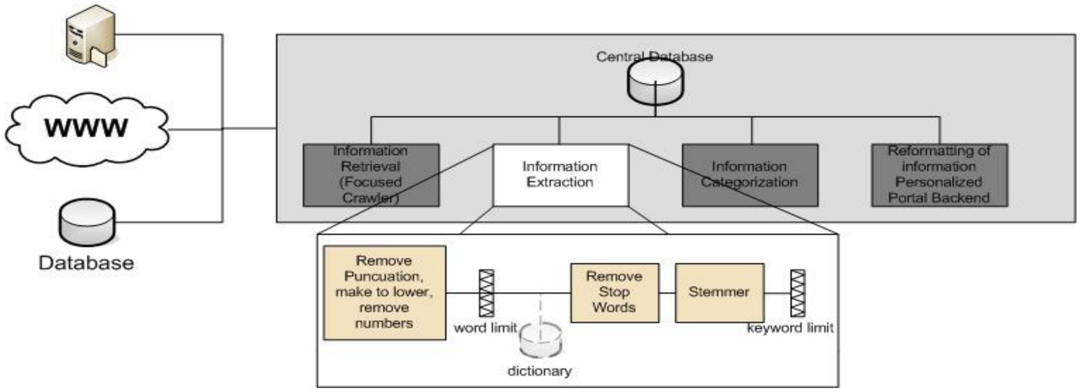
Central Database
WWW
Database
Information Retrieval (Focused Crawler)
Information Extraction
Information Categorization
Reformatting of information Personalized Portal Backend
Remove Puncuation, make to lower, remove numbers
word limit
dictionary
Remove Stop Words
Stemmer
keyword limit

Figure 1: General Architecture of the mechanism

The output of the information extraction mechanism can be multiple depending on the procedure that we follow. If the mechanism is in procedure of capturing pages from the web, extracting the text, processing the text and so on then the mechanism stores the results to the database in order to produce input for the mechanisms that follow. If we try to make our experimental procedure then the output can be either xml or text files with the results.

3. The Structure of Information

Information exists on the net in many types and many forms. Our system specializes on pure text preprocessing. But even the text on the web can be found in many types and the system should be able to treat different types of information with a way that would be appropriate for each type. In our system we will focus on three different kinds of information and we will use the results of the preprocessing procedure in order to explain how the extracted information can be used in order to support different mechanisms.

The first type of information that we will process is papers from publications. Papers are usually more than 6 pages long and they contain more than 3 thousand words. More specifically, papers include formal language and the words are very long and concerning a specific scientific field. The aforementioned mean that when we want to extract information from papers we should keep only the long words and also any words or phrases that are written in capital letters due to the fact that a scientific paper includes specific terms. Keeping short words (shorter than 4 letters) should be proven to be inefficient. In our experiments we will use word length limit of 5, 6 and 7 and more.

The second type of information that we will process is e-mails. E-mails are written in a very informal language. People write their e-mails as if they were talking with the receiver of the “letter”. The words are usually short and without special meaning in terms of scientific fields. When we will process this kind of information we will pay attention to keep even small words as they may include important information. In our experiments we will use word length limit of 3, 4 and 5 and more as we believe that it is enough to include all the meaning of the e-mail.

The last type of information that we will process is articles. Articles are extracted from news portals and are written both in a formal and informal way. This means that it is a text that is in the median of the two aforementioned types like it is shown in the following figure. The amount of text is neither too short nor too large. In this type of information we concluded that we should use word length limit of 4, 5 and 6 and more.

There is always a trade-off with this procedure. If the word length limit is too small then we can be sure that by keeping much information we will produce a list of keywords that contain the full meaning of the text, but we need more space to store them. On the other hand if we have a large word length limit then we produce fewer keywords (than before), which require less space to be stored but we cannot be 100% assured that the remaining keywords include the full meaning of the document.

4. Experimenting With the Information Extraction Mechanism

In this chapter we will present the results of our mechanism. The input to the mechanism would be e-mails, articles and papers. For each one of these types of text we will present the results of preprocessing that leads to keyword extraction and we will explain how we can lessen the keywords without losing the meaning of the text in two ways:

- by keeping words with length limit,
- by keeping a percentage of the keywords produced.

In order to “calculate” the difference in meaning between two texts (ie the one with word limit of 4 and the one with word limit of 6) we will use a simple version of SVM algorithm [18]. If we assume that we have a vector with all the keywords and their frequencies for text A and vector b for text B then we calculate the relevance between the two texts as:

$$x = a*b \tag{1}$$

$$y = |a|*|b| \tag{2}$$

$$z = x / y \tag{3}$$

$$r = sin(z) \tag{4}$$

where x is the dot product of vector A and vector B and y is the product of norm (2) of A and norm (2) of B.

As we can see from the above fraction of code r is limited between the values zero and one. When r is zero then vectors a and b are completely irrelevant and if it is one then the vectors are identical. This means that when “r” is close to one then we have much relevance between the texts that are represented through the vector.

In order to reduce even more the keywords of a text we decided to keep only a percentage of them and recalculate with the above formula the relevance between the keywords from the starting text and the percentage of keywords we have kept.

4.1 Experimenting with e-mails

In this section we will see the results of the mechanism when it processes e-mails. As we said earlier in our paper we will use length limit of 3, 4 and 5 and more characters. In the following chart we can see the results of our experiments.

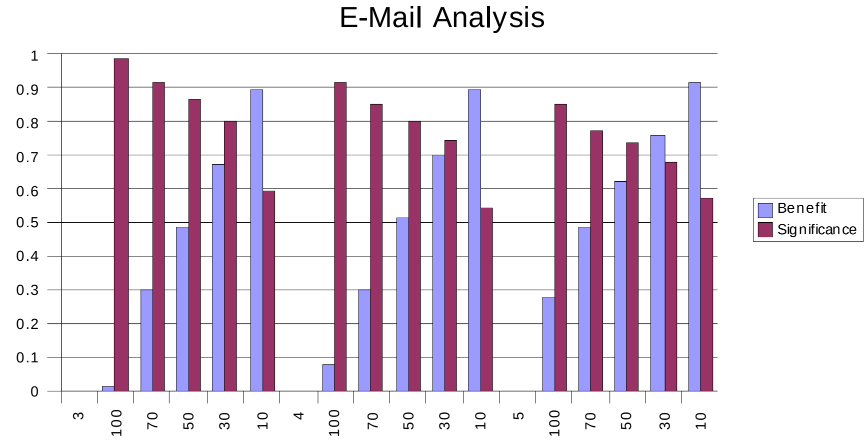

Figure 2: E-mail Analysis

As we can see from the chart we have done a word length limitation of 3, 4 and 5 and more characters and then we have kept a percentage for each of word length limitation. By limiting the word length to 3 and by keeping a 70% percentage of the keywords extracted we assume to have benefit of about 30% of the keywords of the starting text and the significance of the two text is over 90%.

What we care about is the significance between the starting text and the derived keywords. So, we decided to set the limit to the significance to 85% as it was obvious that the keywords remaining were representative to the starting text. The above limitation means that the pairs of word length limit and keyword percentage kept that derive from the chart could be 3/100%, 3/70%, 3/50%, 4/100%, 4/70% and 5/100%. The benefit for each of the above pairs is 1%, 29%, 48%, 8%, 30% and 28% respectively. The ratio of benefit/significance is 0.01, 0.33, 0.56, 0.09, 0.35 and 0.33 for each of the aforementioned pairs. This means that the best pair seems to be 3/50% for the e-mail analysis. We limit the word length to 3 and keep the half of the keywords that derive from the analysis. We have to note that the keywords are order by descending frequency before keeping the correspondent percentage.

4.2 Experimenting with papers

In this section we will see the results of the mechanism when it processes papers. As we said earlier in our paper we will use length limit of 5, 6 and 7 and more characters. In the following chart we can see the results of our experiments.

As we can see from the chart we have done a word length limitation of 5, 6 and 7 and more characters and then we have kept a percentage of the keywords for each of word length limitation. As we can see from the chart the results are not affected by the percentage parameter. This can be explained as follows. The documents that we have include more than 900 unique words which appear many times in the text because the papers are about a specific thematic field and repetition of the keywords is inevitable. In order to find which is the limit of significance that one can admit that the text has not lost its meaning, we will not be strict. The papers are focused in a specific thematic field and it is unusual to find a very general paper. The limit that we chose was 80%. This means that the limitation to word characters to 7 and more seems to be unsuccessful. Keeping that in mind we can see that we have all the categories of 5 and 6 (word character limitation) passing the limit of 80% in significance.

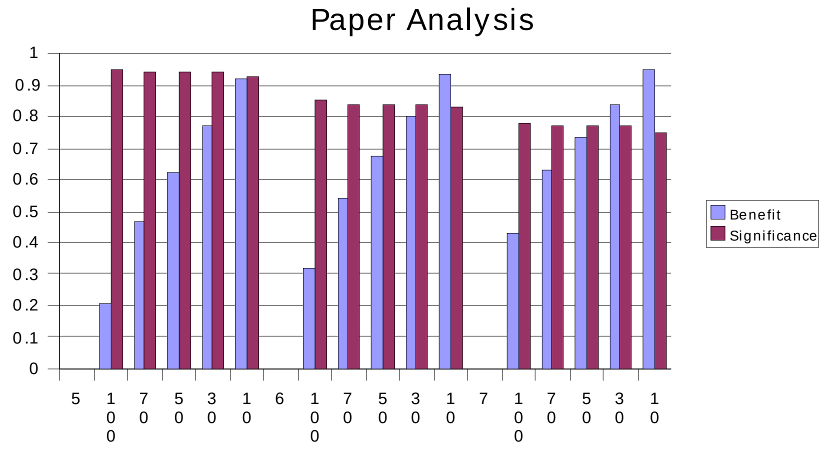

Figure 3: Analysis of Research Publications

As we already mentioned categories of 5 and 6 seem to "pass" and the best of them according to the benefit is subcategory 6/10%. This means 6 characters to word limitation and storing ten percent of the keywords extracted, which lead to 83% significance and more than 90% benefit.

In section 2 we have described that the mechanism that we will present is part of a complete mechanism for information retrieval, extraction and categorization. As we said earlier the limitation to word characters to 7 seems to be unsuccessful, but when we tried to make the whole system run and try to categorize papers with 7 characters and more limitation to words the categorization worked perfectly.

In conclusion we have to note that if we think this mechanism as an independent system then we can say that the pair 6/10% has the best results, but if we think it as part of our mechanism then the pair 7/10% is the best. More analysis on this will be presented in the next section.

4.3 Experimenting with articles

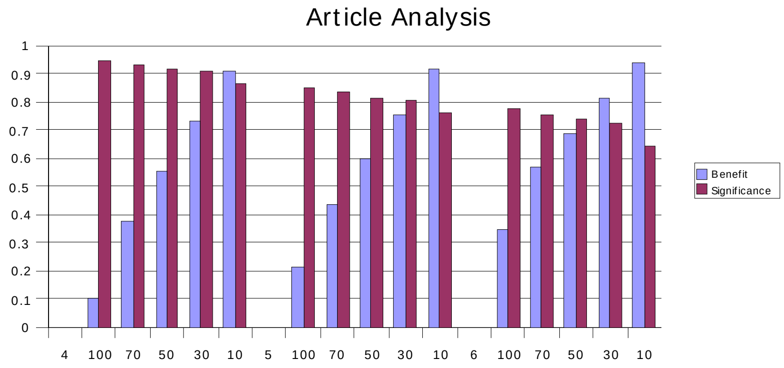

Figure 4: Analysis on Articles collected from news portals

In this section we will present the results from article analysis. As we already mentioned we will use word length limit of 4, 5 and 6 and more characters and we will keep sample percentage of the output keywords in order to find the best pair of word length limit / percentage of keywords that has good results on benefit and significance.

We have to set a limit to the significance that has the best results. This limit is selected by running all the procedures of our system (not only the IE mechanism). For the articles we have selected the limit of 85% in significance. It seems that lowering this limit will lead to unexpected results and by making it 90% or more we face problems of storing the keywords (too many).

The pairs that pass this 85% limit can be found only in categories of 4 and 5 (word length limit). More specifically the pairs that derive from the graph include all the "4" category and the first option of the "5" category. The first option of the "5" category has very low benefit (21%). This means that this option will be excluded and the remaining are from category "4" and more specifically the pair 4/10%. In this pair we obviously have more than 85% significance and more than 90%

benefit. This means that if we cut 90% of the unique keywords and store only a 10%, the remaining ones are more than 85% significant to the starting keywords according to the algorithm described in the first paragraph of this section. This can be acceptable if we realize how much information we have removed from the text.

4.4 General Results

After experimenting with different types of text we can obviously notice that different types of text need dissimilar treatment from the preprocessing mechanism. As it was denoted in section 3 the plain text of an e-mail is extremely different from the plain text of a publication. As we can see from the results, in the e-mails we should keep all the words with small length limit and create a stop word list with few words. On the other hand we should be careful when treating a publication. The words used are formal and usually very long. We can benefit from this assumption by setting the words' length limit to a higher level and keeping a quite small percentage of the keywords in order to “represent” the text. In addition the stop word list must be very long and strict.

We expected that gaining in significance of the final keyword list to the primary keyword list will lead to lessening of the benefit. Though we managed to keep a high percentage for both of them. This means that we managed for all the different types of text to conclude to a keywords list that is more than 80% smaller than the starting one and more than 80% significant to the starting one. In more words, when we have a text with 5000 words, we believe that by keeping less than 20% of them (100 words) we can have a very good “representation” of the text.

Moreover, what we can extract from the result is that there is no need to index a text using all the terms included. This means that we can store only few parts of a text and not the whole text (indexed) lowering in this the high needs for storage. We believe that gaining more space for the storing needs of a system is the wrong answer to the storage problem. What we can do is just throw away the useless information. From this derives that the natural language includes a high percent of what is called “garbage”, not only useless but unwanted information.

Concluding what we extracted from our results can be gravitated to the three points: (1) Information Extraction Systems have to focus on the type of information that they are accessing and adapt on them, (2) the text documents to be processed include useless information that must not be processed but deleted and (3) there is no need to store a full text for indexing but according to our results less than 20% of its text is enough.

5. Future Work – Conclusion

In the paper we have presented a part of system for information retrieval, extraction categorization which leads to the creation of a personalized portal. We have presented the results of the experiments made for the information extraction mechanism and especially for the preprocessing procedure, which is fully parameterized. We have paid attention to issues that have to do with the input and output of the system in order to make it autonomous. We have also analyzed the reasons, why the system has to be parameterized according to the difference of the texts that it processes. The results had to do with the difference in parameters of word length limit, stop word lists, the stemmer and the percentage of keywords that we decide to keep. We have concluded to different pairs of word length limit / percentage of keywords that suits better each type of text concerning e-mails, articles and papers from conferences. We concluded that in texts that are similar to the natural daily speaking language (e-mails) we have to keep in low levels the word length limit and maintain a high percentage of the keywords extracted. On the other hand the papers include formal language and the experiments showed that keeping the word length limit in higher levels (6 or 7 characters) and the percentage of the keywords extracted to a low level (10% of the keywords extracted – from 90% to 100% of the keywords ordered by frequency ascending) we have great results.

We believe that the mechanism is in very premature level. Many changes can be proposed in order to make the system more efficient. Some of them can be the addition of a dictionary. In section 2 (Architecture) we can see that the system includes a dictionary. This can be very helpful in order to correct the grammatical mistakes that can be found in a text. The stop word list and the stemming algorithm would work much better and the results could be more accurate. Moreover, we can add a feature that would calculate differently the weight of the keywords in a text and not just the frequency. A very simple but efficient addition could be to count three or more times more the frequency of the keywords that are found in the subject, because the subject of a mail, the title of an article or a paper usually has much to tell about the meaning of the text. The same can be assumed for the first sentence of a paragraph, or for the title of a chapter.

More additions can be made to the mechanism in order to make him “understand” the type of the text the it is processing and give the appropriate value to it parameters without having someone to do it. This can be simply done by counting the words of a text. A more “clever” way to do this is by keeping statistics about the texts that are being processed. As we can see from the experiments that we have done, the types of text have different values for the significance and benefit. When the system will process an e-mail it could give results similar to the ones presented in chapter 5. This means that we can realize from the results the type of the text. Making the mechanism more automatic and more autonomous is our intention. Our goal is to create an information extraction mechanism that will be able to process many kinds of inputs, realize the type of text, its subject (difference in stop word list), and understand the percentage of the keywords that has to be stored.

The use of this mechanism, as already mentioned, is its support for information extraction and information categorization mechanisms. This keyword extraction mechanism is actually a module of a complete mechanism, starting from information retrieval and finalizing in a personalized portal. In our occasion this module is used in order to support a text summarization mechanism which leads – with the help of the Keyword Extraction module – to text categorization. More specifically, we use the extracted keywords in order to determine the category of our retrieved texts and additionally create a summary of the text in order to present to the users a part of the text and not the whole text.

References

[1] Adams, K., The Web as Database: New Extraction Technologies and Content Management, http://www.onlinemag.net/OL2001/adams3_01.html ONLINE, 2001

[2]Text Mining with Information Extraction Un Yong Nahm

[3] XML, T Bray, J Paoli, CM Sperberg - McQueen - *W3C Recommendation*, 2000 - xml.coverpages.org

[4] Cynthia A. Thompson, Joseph Smarr, Huy Nguyen, Christopher Manning, Finding Educational Resources on the Web: Exploiting Automatic Extraction of Metadata, *Workshop on Adaptive Text Extraction and Mining*, Croatia, 2003.

[5] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale Information Extraction in KnowItAll. *In Proceedings of the 13th International World-Wide Web Conference*, 2004.

[6] Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun Kumar Ramani, and Yuk Wah Wong. Comparative Experiments on Learning Information Extractors for Proteins and their Interactions. A*rtificial Intelligence in Medicine (Special Issue on Summarization and Information Extraction from Medical Documents)*, 33, 2 (2005).

[7] David Pinto, Andrew McCallum, Xing Wei, W. Bruce Croft. Table extraction using conditional random fields. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, 2003

[8] R Dale, H Moisl, HL Somers, Handbook of natural language processing, 2000

[9] Gaizauskas, Robert and Yorick Wilks. "Information Extraction: Beyond Document Retrieval." J*ournal of Documentation*. 54, no. 1 (January 1998)

[10] N Kushmerick, B Thomas. Adaptive Information Extraction: Core Technologies for Information Agents. *AgentLink*, 2003

[11] Mori, J., Matsuo, Y., Ishizuka, M., Faltings, B. Keyword Extraction from the Web for Personal Metadata Annotation, 2003

[12] Tonella, P., Ricca, F., Pianta, E. And Girardi, Ch. Using Keyword Extraction for Web Site Clustering. In Proceedings of the Fifth IEEE International Workshop on Web Site Evolution (WSE03), 2003.

[13] Anette Hulth. Improved Automatic Keyword Extraction Given More Linguistic Knowledge. Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, 2003.

[14] C. Bouras, G. Kounenis, I. Misedakis, V. Poulopoulos. A Web Clipping Service’s Information Extraction Mechanism. 3rd International Conference on Universal Access in Human - Computer Interaction, Las Vegas, Nevada, USA, 2005

[15] I. Antonellis, C. Bouras, V. Poulopoulos, A. Zouzias. Scalability of Text Classification. (To be Presented) in International Conference on Web Information Systems and Technologies (WEBIST06), 2006.

[16] Martin Porter, Porter Stemmer. www.tartarus.org/~martin/PorterStemmer/

[17] Chris Paice and Gareth Husk. Paice/Husk Stemming Algorithm. http://www.comp.lancs.ac.uk/computing/research/stemming/Links/paice.htm

[18] Vapnik, V. Statistical Learning Theory. Wiley-Interscience, New York, (1998).