Sobek: a Text Mining Tool for Educational Applications

[1]E. Reategui, [1]M. Klemann, [2]D. Epstein, [3]A. Lorenzatti
[1]PPGEDU/PPGIE - UFRGS, Av. Paulo Gama, 110, 90040-060, Porto Alegre, RS - Brazil,
eliseoreategui@gmail.com, miriamklemann@gmail.com
[2]Informatics Institute - UFRGS, Postal box 15064, 91501-970 Porto Alegre, RS, Brazil
daepstein@gmail.com
[3]Endeeper, Av. Bento Gonçalves, 9500 - UFRGS, 91509-900 - Porto Alegre - RS – Brazil
alorenza@gmail.com

Abstract — *This paper presents a mining tool to extract relevant terms and relationships from texts, and proposes its use in educational applications. A particular text mining technique is employed to analyze texts and build graphs from them, in which nodes represent concepts and edges represent the relationships between them. Some adjustments are proposed here in the original mining and representation methods, in order to provide results which are more suitable for our educational applications. Two experiments exemplifying the extraction of graphs from students' essays are presented in the paper. Results showed that the mining tool was able to identify a considerable number of relevant terms from the texts analyzed, providing concise representations of documents which can support students' and teachers' tasks.*

Keywords: text mining, graphs, education

1. Introduction

In recent years, data mining and text mining have become more popular in the field of Education mostly because of the growing number of systems which store large databases about students, their accesses to material available, their assignments and corresponding grades. Such expansion in the field yielded the establishment of a community committed to Educational Data Mining applications. This community is concerned mostly with the development of methods for exploring data coming from educational settings, and employing those methods to better understand students and learning processes [2]. In this work, our main goal has been to design and develop a text mining tool to be used in educational applications. Below, we list a few examples of the uses of the tool to support either students' or teachers' work:

- Helping teachers to evaluate students writings from a qualitative point of view;
- Assisting teachers in identifying the significance of students' contributions in discussion forums;
- Supporting reading strategies;
- Supporting text writing;

A particular text mining technique based on statistical analysis has been used to extract graphs from texts, representing relevant terms and their relationships [1]. This technique has been customized here in order to provide results which were more suitable for the targeted applications. Typically, for long documents, the original mining algorithm extracted graphs that were too large to be comprehensible in the proposed educational applications. The changes implemented worked on the reduction of the number of nodes and relationships of the graphs, including on the extraction of compound terms. The next section presents different methods for representing data extracted from texts, including graph-based approaches. Section 3 describes the text mining method on which we have based our research, detailing what has been changed in the original algorithms. Section 4 presents the text mining tool Sobek, and section 5 describes some experiments carried out in order to validate this research. The last section of the paper presents conclusions are directions for future work.

2. Representing information extracted from texts

Representing the information extracted from texts requires specific data-structures. Graphs are an interesting approach which can be used to organize words extracted from texts and keep the relationships between them, including their location inside the text. Since graphs are an abstraction created to represent relationships between objects or concepts, they are easily understood and are widely applied [3]. Adam Schenker proposed a text mining technique to extract information from Internet pages, and defined six different graph models to represent the information extracted from the texts [1]. One of these models, the n-simple distance model, is based on the idea that each statistically relevant word of the text should be connected to the N subsequent relevant words. An interesting feature of this representation approach is that it enables the storage of the relationships between relevant terms found in a text.

Another common representation scheme which has been frequently used in Information Retrieval systems is the vector space model, typically used in text retrieval and document ranking [4][5]. Different adaptations in the model have been proposed along the years as to adjust it to very distinct applications, such as content-based image retrieval combining textual and visual data [6], user modeling [7] or web information retrieval [8]. One of the main features of the

model is to represent each possible term that can appear in a document as a feature dimension. The value assigned to each dimension indicates the number of times the corresponding term appears on it, or it may be a weight that takes into account other frequency information, such as the number of documents in which the terms appear. The model is simple and allows the use of traditional machine learning methods that deal with numerical feature vectors in a Euclidean feature space. However, it discards information such as the order in which the terms appear, where in the document the terms are, how close the terms are to each other, and so forth. As in our educational application it was important to keep a more precise representation regarding the relationships between terms, the n-simple distance model seemed to be a more suitable alternative. The next section explains and proposes some small changes in the model.

3. The Text Mining Method

The text mining method used in this work has been based on the n-simple distance graph model, in which nodes represent the main terms found in the text, and the edges used to link nodes represent adjacency information [1]. Therefore, nodes and edges represent how the terms appear together in the text. Figure 1 shows a graph extracted from a short text about the atomic bomb.

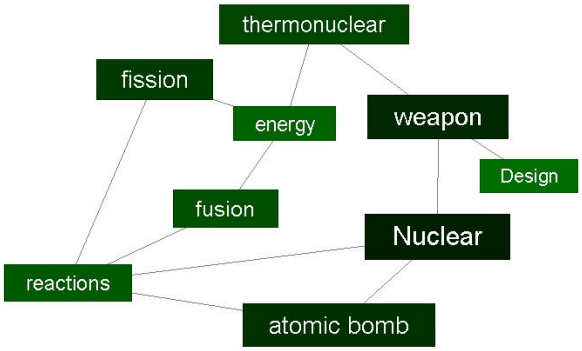

Fig. 1. Graph representing relevant terms extracted from a text about the atomic bomb.

In our graphical representation of the graph, nodes which are more relevant are presented in a larger rectangle and in darker color (e.g. the terms "Nuclear", "weapon", "atomic bomb"). While other text mining approaches rely on the analysis of relevant morph syntactic patterns (such as *"Noun Noun"*, *"Noun Preposition Noun"*, *"Adjective Noun"*, etc.) in order to generate compound terms for the mining process [9], here we used a simpler method which was based on the frequency with which these compound terms appeared in the text.

The method used here relies on a parameter *n* to extract the compound concepts with more than one word. According to this parameter we create a combination of the current word with the *n* subsequent words. What we try to do is to create a wide combination of words to find the most frequent group of words that appear in the text. For instance, considering *n =3*, the analysis of the sequence of terms "AA BB CC DD EE FF GG HH" would lead us to the following combinations "AA", "AA BB", "AA BB CC", "BB", "BB CC", "BB CC DD", and so on. In order to avoid sequences starting with prepositions or articles, specific filters are used. After identifying the most frequent combinations of words, which we will call *concepts*, the mining process selects the primary set of relevant ones based on their frequency in the text.

The next step is to compute the similarity between *concepts*. Consider two *concepts* x = "AA DD BB" and z = "BB CC DD EE FF AA". The similarity coefficient between them is computed with the dot product also used in the Vector Space Model.

The similarity coefficient, represented by S, computes the quantity of words present in both concepts represented by P, and the number of words of the larger concept represented by B. Therefore we have:

$$S = P / B$$

In the example above $S=0.5$ as the concepts have three words in common, words "AA", "BB" and "DD". Concept z, being the biggest, has six terms. After computing the value of S, the relevancy coefficient R is computed for each concept. The number of words of the concept (C) and the absolute frequency (F) are introduced in the computation process. To calculate the R value for each concept, the following formula is employed:

$$R = S * C + F$$

The concept with the largest value for R is kept on the base, and at the end of the process, it is included in the graph. In the example above, let us consider that concept x has $C=3$ and $F=3$, and concept y has $C=6$ and $F=2$. We can conclude that concept z is going to remain in the base to be part of the graph, even if its F value is smaller then that of concept x. The idea behind the relevancy coefficient R is to compare two concepts and keep the one that "says mores", even if it appears a fewer number of times.

4. The Text Mining Tool Sobek

The text mining tool Sobek was developed using the method described in the previous section. The name Sobek comes from the Egyptian mythology where it represents a god related with discernment and patience, features needed to find out relevant and useful information from large amounts of data. Sobek was developed using the Java programming language. The Interactive Graph Drawing API was used to render the graphs on the screen [10]. Sobek is able to analyze documents in "TXT" format, as well as in "PDF" and "DOC" formats. This functionality has been obtained with the use of two different APIs: JPedal [11] and POI [12].

Although Sobek can be employed for the analysis of any type of text, its development has been originally inspired by the need of university teachers who work with distant learning and who have to read dozens of texts, messages and posts written by students [13]. By providing these teachers

with concise graphical representations of the students' texts, Sobek enables them to speed up their work, giving these educators more time to concentrate on specific problems which have to be tackled.

Sobek can be used in different ways. The analysis of plain text is Sobek's simplest operation. The text to be analyzed can be copied and pasted in the tool or it can be loaded from a file. If the text is in a "PDF" or "DOC" format, it is automatically converted to the text format. The main goal of the text analysis is to extract relevant terms and concepts from the text and to visualize their graphical representation in the form of a graph. A teacher could use this procedure, for instance, to visualize and get the main ideas students addressed in their essays. The interface of the mining tool is presented in figure 2, where a text about the atomic bomb has been loaded. The resulting graph obtained from the mining process has been shown in a previous example (figure 1).

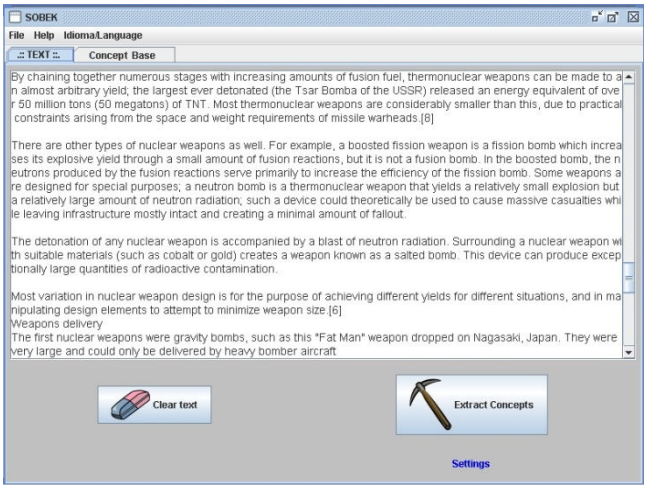

Fig. 2. Sobek's main graphical user interface.

Sobek can also analyze a collection of texts, even if they are in different formats. By comparing the list of terms extracted from the collection of texts with the terms extracted from a student's essay, the system may help teachers to check whether the student's essay addressed topics which were important to be covered. For instance, a teacher could ask his/her students to write a paper based on other articles and texts. Using Sobek, the bibliography given to the students could be employed to create a base of concepts. This database may then be used to verify if the students discussed in their articles the main issues contained in the literature suggested. The base of concepts created automatically can also be edited. Concepts can be added or removed and new relationships between them can be created or extinguished. The base of concepts does not necessarily have to be created from a collection of articles; it can also be created manually by adding concepts and establishing their relationships.

Sobek's first step is to break one or more texts into a set P of words w. This set P of words is analyzed statistically so we may know how many times each word appears in the texts. During the extraction of the words a list of stopwords is used to remove articles, prepositions and terms with no meaning from the base of concepts.

In order to narrow down the number of concepts on the graph and keep only the most important words, we propose the following method:

- Firstly, we set a minimum frequency Θ that will indicate the lowest number of occurrences that a word w must have in order to appear in the graph. As our goal is to provide students and teachers with a concise representation about a text, or a group of texts, it is important to present them only with the most relevant information. Thus, we discard those terms that have a number of occurrences lower than Θ, producing a smaller graph without too many irrelevant terms.
- Secondly, we use a stemmer to remove inflectional and derivational endings of words in order to reduce word forms to a common stem. For example, there is no need for both words *lamp* and *lamps* to appear in the graph. As they both express the same meaning, the one with the highest number of occurrences is displayed.
- After Sobek removes from set P those words that will not be part of the graph, it must verify the relationship between the ones remaining. To do so, for each word $w_i \in P$, we must know which words $w_{j,k} \in P$ come after and before it in the original text. The terms w_j and w_k, called "neighbors of w_j", are added to a list N_i with a counter. If a word w_j appears more than once after or before the word w_i in the original text, its counter will be increased in list N_i. After this process is completed, we have a list of every concept and its neighbors and we can sort it to use only those neighbors with the highest counters. This process enables the tool to display only important relationships between terms, being based on the idea that if word w_j is the neighbor with the highest counter for w_i, it is likely that those two words have some meaning together.

To determine how many relationships have to be shown in the graph for each concept, we use the number of occurrences for those concepts. As a fully connected graph provides no information about how each word is related to the next, we set a maximum number of possible connections Ω. The number of connections for a word $w_i \in P$, will be called Con_i here. The number of occurrence of the word w_i in the original text will be called $NumOc_i$ and the highest number of occurrence will be called $MaxOc$. The word with the highest number of occurrence will have, at most, Ω connections in the graph. Each word will have a number of connections proportional to its $NumOc_i$ and to $MaxOc$. Hence, the number of connections for each word $w_i \in P$ is:

$$Con_i = \frac{NumOc_i * \Omega}{MaxOc}$$

This will assure that those concepts that have a higher $NumOc_i$ will also have a higher Con_i, as they seem to be more important concepts in the text.

5. Evaluation and Results

A first experiment was carried out in order to verify how accurate were the graphs extracted by Sobek. Initially, a two-pages text[1] (816 words) about the topic "Realism" was presented to 20 high school students. The students extracted the graphs from the text, and then each of them wrote about the topic, being able to use the original text as well as the graph as a reference. Figure 3 shows the graph obtained from the text used in the experiment.

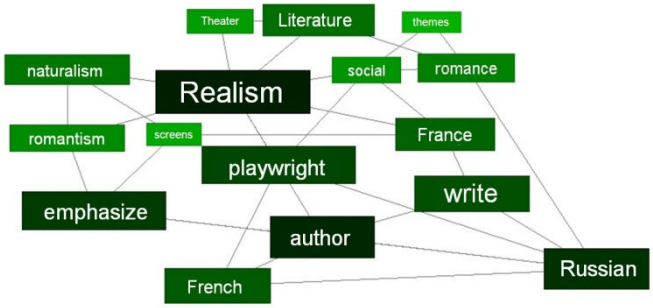

Fig. 3. Graph extracted from text about Realism

The essays produced by the students were then analysed to verify whether the terms of the graph were also present in the students' writings. The results showed that each student used, on average, between nine and ten terms of the graph in his/her essay (9.85 terms, to be precise). Considering that the graph contained a total of 16 terms, an average of 61.6% of the terms identified by the tool as relevant appeared in each text produced. Such results demonstrate that Sobek was able emphasize a considerable number of relevant terms from the text. Table 1 shows the total occurrence of each of the graph's terms in all of the students' texts.

The most cited term of the graph in the students' texts was the word "Realism", which is also highlighted in the graph as the most important term. The least cited terms were the words "write" and "France". Although the order of importance given in the graph for all of the terms does not follow exactly the number of occurrence of these terms in the students' writings, it is interesting to observe that all of the terms extracted from the original text were used by the students in their essays. Such results reinforce the fact that the mining algorithm was able highlight a large number of relevant terms from the text.

A complementary experiment was carried out with Sobek in order to evaluate its capacity to provide summary representations of students' writings. Seven undergraduate students in Computer Science related courses participated, being asked to discuss in a forum about how to design websites, the tools and programming languages available, and the artistic abilities involved.

[1] Complete text, in Portuguese, available at: http://www.artesbr.hpg.ig.com.br/Educacao/11/interna_hpg10.html

Table 1: Total occurrence of terms in students' texts

Graph's terms	Number of occurrence
Realism	100
author	34
Russian	9
playwright	15
emphasize	12
write	5
Literature	42
France	5
romantism	24
naturalism	24
romance	18
Theater	34
social	10
theme	23
screen	23

The teacher who proposed the activity confirmed that the graphs extracted from the students' essays provided a good way to grasp the main topics discussed. Furthermore, the graphs elicited automatically were said to be of great "help not to understand thoroughly what the students had to say, but mostly to skim through good and bad contributions". Figure 4 shows one of the graphs extracted from a student's essay[2].

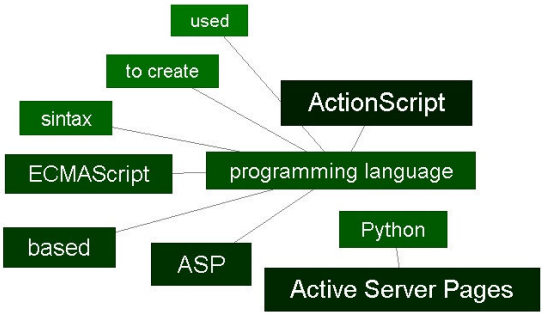

Fig. 4. Graph extracted from a student's essay.

In another example, we may observe how a concise representation of a student's post can reveal a non significant contribution (figure 5).

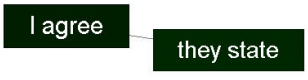

Fig. 5. Smaller graph from another student's writings.

In the example, the only idea represented in the graph was that the student agreed with what was said by other

[2] The graphs presented in this section were originally created in Portuguese, but they have been translated into English here to improve the readability of the paper.

classmates. These results reiterate the applicability of Sobek's mining procedures to elicit relevant information from texts.

6. Discussion

This paper focused on detailing Sobek's main algorithm and showing its effectiveness in providing correct representations of texts' contents, without dealing with domain-specific knowledge. Gao et al. [14] also proposed a method for extracting terms from texts automatically, focusing mainly in business applications. Our approach to differs considerably from this method mainly for its representation mechanism based on graphs, and the consequent specificity it its algorithms.

Other approaches rely heavily on domain knowledge to enrich the mining process, as in Dayanik [15] where a method for text classification is proposed. Information extraction mechanisms also employ domain-specific knowledge to tag and classify data from unstructured or semi-structured documents [16]. The benefits of using or not using domain-specific knowledge in mining applications has also been compared [17], showing advantages and disadvantages of each approach. In the case of our educational applications, it has been an important feature that the mining method runs automatically, with none or little interference from students or teachers. Because of that, the use of domain-specific knowledge has not been considered yet. However, we are currently working on the integration of Sobek with domain ontologies and natural language processing, in order to tag terms and obtain more refined results in the mining process.

As for the presentation of the mining results, other tools present relevant terms extracted from texts by highlighting these terms in the actual document [18], or by simply ranking terms through a frequency count [19][20]. Our solution is based on a more visual representation. From an educational perspective, presenting the mining results in the form of a graph is interesting as it takes learners to focus of concepts and their relationships, and to reflect about them. The associations suggested by the graphs lead learners to reasoning processes that are in many respects similar to those which are triggered during the analysis/development of conceptual maps [21]. However, while conceptual maps represent *noun verb noun* propositions, our representation do not follow such norm. In our mining method, terms may be connected without following any syntatic rule. Besides, conceptual maps are normally built manually, do not relying on any automatic mechanism.

Information Retrieval (IR) is another approach which can be used to find relevant information on textual data, providing an easy way to search textual documents by indexing them with a collection of words. While IR's main objective is to find the right information to satisfy a given query [22], our goal has been to look for hidden patterns inside a body of textual data, with the focus of providing concise representations of documents.

7. Conclusion

The main contribution of this work has been to design a text mining mechanism for educational applications where we proposed a modification in a known text mining process as to produce more knowledgeable outcomes. While the original method generated graphs with one single term represented in each node and with a very large number of connections, in our approach several terms can be placed in a single graph node, and the number of connections have been considerably reduced, according to the size of the text being mined. It could be argued that by connecting nodes with words that appear together frequently in the text, one could represent concepts just the same way we do by placing them together in a single node. However, for the user who has to interpret the graph, it is more difficult to grasp the meaning of a compound term that is dispersed in different nodes.

Other known text mining methods group together terms in order to make more accurate concept extraction from texts, as in [9] where relevant morph syntactic patterns are searched for in order to create meaningful tokens. While such procedure relies on the some level of linguistic processing, our approach is much simpler in that it is based mainly on a statistical analysis of the frequency with which the complete tokens appear in the texts.

Previous research has already shown promising results regarding the use of Sobek in educational applications. For instance, in Macedo et al. [13] it has been demonstrated how Sobek and its graph representation mechanism could give teachers a concise view of the students' assignments by emphasizing important concepts that appeared in the texts. The results of the experiments carried out demonstrated the potential of Sobek's text mining for the analysis of students' work. Furthermore, the tool has also been evaluated by Azevedo et al. [23], who proposed a method for identifying the quality of contributions in discussion forums through a computational method employing the graphs extracted from the students' posts. The authors showed how it is possible to order students' contributions through their concise representations extracted by Sobek. Here, we have focused on detailing Sobek's mining process, specially the add-ons made in the original method. Our validation procedures emphasized not so much the applicability of the mining tool in educational settings, but the accuracy of the mining results.

As for the current use of Sobek, the possibility to create a database of concepts before mining students' contributions showed to be useful approach when dealing with small texts. As discussed in [24], it has been observed that the simple application of statistical analysis on small texts can produce undesirable results, which is inevitable. Sobek is currently being integrated to a virtual learning environment and it will be used by a large number of teachers in several courses. The tool's mining feature is also being improved by connecting it to different domain ontologies.

Acknowledgment

This work has been partially supported by the National Council for Scientific and Technological Development (CNPq - Brazil) under grant 476398/2010-0, FAPERGS Research Support Foundation, under grant 1018248, and Fiocruz-Fiotec project no. ENSP 060 LIV 09.

References

[1] A. Schenker. “Graph-Theoretic Techniques for Web Content Mining”. PhD thesis, University of South Florida, 2003.

[2] R. S. J. D. Baker, K. Yacef. “The State of Educational Data Mining in 2009: A Review and Future Visions”, Journal of Educational Data Mining, vol. 1, no. 1, p. 3-17, Oct. 2009.

[3] M. Chein, M-L. Mugnier. “Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs”, Berlin: Springer Verlag, 2009.

[4] V. V. Raghavan, S. K. M. Wong. “A critical analysis of vector space model for information retrieval“. Journal of the American Society for Information Science, vol. 37, no. 5, John Wiley & Sons, p. 279-287, 1986.

[5] D. L. Lee, H. Chuang, K. Seamons, “Document Ranking and the Vector-Space Model,” IEEE Software, vol. 14, no. 2, p. 67-75, Mar./Apr. 1997.

[6] T. Berber, A. Alpkocak. “An extended vector space model for content-based image retrieval”. In Proceedings of the 10th international conference on Cross-language evaluation forum: multimedia experiments (CLEF'09), C. Peters, B. Caputo, J. Gonzalo, G. J. F. Jones, and J. Kalpathy-Cramer (Eds.). Berlin: Springer-Verlag, p. 219-222, 2009.

[7] E. Mangina, J. Kilbride. “Utilizing vector space models for user modeling within e-learning environments”. Computers in Education, vol. *51,* no. 2, p. 493-505, September 2008.

[8] W. Luo, C. Liu, Z. Liu, C. Wang. “On N-layer Vector Space Model-Based Web Information Retrieval”. In Proceedings of 6th the International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Chengdu, China, 23-25 September, p. 1 – 3, 2010.

[9] R. Feldman, M. Fresko, Y. Kinar, Y. Lindell, O. Liphstat, M. Rajman, Y. Schler, and O. Zamir. “Text mining at the term level”. In Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD '98), J. M. Zytkow and M. Quafafou, Eds., London, UK: Springer-Verlag, p. 65-73, 1998.

[10] U. Erlingsson and M. Krishnamoorthy. Interactive Graph Drawing. Available at: http://www.cs.rpi.edu/research/groups/pb/graphdraw/. Accessed in March, 2011.

[11] JPedal. Available at: http://www.jpedal.org/ Accessed in March, 2011.

[12] Apache POI Java API. Available at: http://poi.apache.org/ Accessed in March, 2011.

[13] A. Macedo, E. Reategui, A. Lorenzatti, and P. A. Behar. “Using Text-Mining to Support the Evaluation of Texts Produced Collaboratively”, in Education and Technology for a Better World: Selected papers of the 9th World Conference on Computers in Education, A. and A. Jones, Eds, Berlin: Springer, 2009, p. 368-377.

[14] X. Gao, S. Murugesan, B. Lo, "Extraction of Keyterms by Simple Text Mining for Business Information Retrieval", In IEEE International Conference on e-Business Engineering, 2005, p.332-339.

[15] A. Dayanik, Using domain knowledge for text mining, PhD dissertation, Rutgers State University, New Brunswick, NJ, 2006.

[16] R. Feldman and J. Sanger. Text Mining Handbook. Cambridge, UK: Cambridge University Press, 2006.

[17] M. Banko, and O. Etzioni. “The Tradeoffs Between Traditional and Open Relation Extraction”, In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, Stroudsburg, PA: Association for Computational Linguistics, p. 600-607, 2006.

[18] K. Frantzi, S. Ananiadou, and H, Mima. “Automatic recognition of multi-word terms”. International Journal of Digital Libraries, vol. 3, no. 2, p.117-132, 2000.

[19] Textanalyser. Available at: http://textalyser.net/ Accessed in March, 2011.

[20] Wordcounter. Available at: http://www.wordcounter.com/ Accessed in March, 2011.

[21] J. D. Novak and D. B. Gowin. Learning how to learn. New York, NY: Cambridge University Press, 1984.

[22] C. D. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge, UK: Cambridge University Press, 2008.

[23] B. Azevedo, E. Reategui, P. Behar. “Qualitative Analysis of Discussion Forums”. In Proceedings of IADIS International Conference on e-Learning, Freiburg, Germany, 2010.

[24] D. S. Leite, L. H. Rino, T. A. Pardo, M. .G. Nunes, “Extractive Automatic Summarization: Does more linguistic knowledge make a difference?”. In Proceedings of the Workshop of Graph-based Algorithms for Natural Language Processing (TextGraphs-2), Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics, Rochester-NY, p. 17-24, 2007.