High performance environment for knowledge discovering in Portuguese language texts in the Web

V. M. Bastos & N. F. F. Ebecken
COPPE – Federal University of Rio de Janeiro, Brazil

Abstract

This paper describes the development and implementation of a practical and efficient methodology to construct a knowledge extraction environment that contemplates the search of information from Portuguese language Web sites. The application includes some text mining facilities, such as similarity and difference identification between pages and sites, content classification and document clustering.

The application conception has its origin on the evaluation environment of competitive intelligence tasks over the Web. The increasing availability of information in the Web has motivated the proposal of an environment that presents the solutions in an integrated form, supplying results analysis according to the user indication.

Keywords Web mining, business applications, knowledge discovering.

1 Introduction

With the increasing availability of information on the Web, the identification and discovery involved in finding useful information became an onerous task that consumes much time. The search for information with added value in great masses of data has become reality. It has become increasingly necessary to have an automatic tool that evaluates this content, bringing to the attention of the consultants, the information, or, the isolated document classification or even though the strategic knowledge of basic importance for decision making.

Text mining tools permit information extraction, which must be understandable, as accurate as possible and surprising. It is possible to understand the information through the knowledge presentation in the form of

WIT Transactions on Information and Communication Technologies, Vol 37, © 2006 WIT Press
www.witpress.com, ISSN 1743-3517 (on-line)
doi:10.2495/DATA060211

association rules [1]. The use of algorithms with embedded criteria produce the most accurate results [2]. The surprise content of documents is the largest challenge in this work, however, it implies in the extraction of a new and interesting knowledge [3].

This work presents a knowledge extraction environment methodology based on the search for inside information in Portuguese language Web sites, that concentrates on some functionalities of text mining in only one tool: classification, similarities, differences and clustering.

2 Methodology

For each method used in the system, there is a specific treatment for the resultant information. But, in all of the cases, the semantic space construction is necessary, extracting relevant terms that are contributing during the results analysis phase.

The Web site pages are pre-processed, and the existing terms in each page are stored in a database, with its stem [4]. Some treatments, such as frequency calculation, are carried through during the site process reading. The selected terms are those on the tags that define the texts in the pages. Although the identified terms such as *stopword* – frequent used terms, such as conjunctions, prepositions and articles, however without semantics relevance – are not considered in the next phase, they are all stored in the database.

The linear (TF) and inverse (IDF) frequencies form the space vector representation of each term and are used in similarity and difference identification processes between pages or sites contents [3]. So, the analyst will be able to identify what is important among the terms used for a comparative site, and in documents clustering process.

Text categorization is the task of deciding whether a document belongs to a set of prespecified classes of documents. Automatic classification schemes can greatly facilitate the process of categorization. Some references describe this task in detail [5, 6], and two methods had been tested with this tool.

3 The development environment

The system is a Web-based application and the development environment is composed for the Windows XP or 2000 operating system, Eclipse platform, Java programming language, Java Server Page Technology, as interface language to create dynamic Web content for this kind of applications development, and SQL Server 2000 as the database management system.

The Eclipse Platform is designed for building integrated development environments (IDEs) that can be used to construct applications as Web sites, embedded JavaTM programs, C++ programs, and Enterprise JavaBeansTM.

The Java programming language [7], also available in [8], was chosen for being a complete programming language, adjusted to the development of Web-based applications, of closed networks or stand-alone programs.

As part of the Java family, technology JSP, available in [9], allows the fast development of independent platform applications. This technology allows the programmer to develop and keep dynamic pages, separating the user interface (presentation layer) of the content and logic application management (business layer), thus making it possible to change the layout with no alteration of dynamic contents.

4 Text mining functionalities

The functionalities developed in this system are: *pre-processing* that reads the Web site content, treating the read terms and saving in a database, *similarities and differences*, finding unexpected information inside the Web site pages, *classification* that involves attributing one or more pre-defined classes to documents, facilitating its automatic search, and *clustering*, that compares pages contents and group together those that are similar.

4.1 Pre-processing

The data entry module executes the pages pre-processing of the selected site, and is used, basically, for reading the information contained in the site pages indicated by the user. This process is called *Web crawler* [10], where the user has the option to select a new URL or read a URL already visited. The obtained information is stored in the database, differentiated by the date and hour (timestamp) the site was accessed.

In this module, all the considered terms as *stopwords* are marked and this information is indicated as a term attribute. The user can also supply the desired *stopwords* list that will be considered in this execution, by text file or by typing the terms in the screen.

The terms are converted for its canonic form, e.g. verbs in imperative form: “estava”, “estou”, estive” to “estar”. And the terms are reduced to its *stem*, through the application of the “stemming” algorithm adapted for the Portuguese language [4], that performs significantly better than the Portuguese Porter algorithm version [11]. The sequence of steps is: plural reduction, feminine reduction, adverb reduction, augmentative/diminutive reduction, noun suffix reduction, verb suffix reduction, vowel removal and accents removal.

4.2 Similarities and Differences

The Similarities and Differences module identifies the similarities and differences between terms of the pages in the same site, between pages of different sites or between sites. The results visualization presents the percentage of similarity or difference between the comparative pages. This functionality allows the user to define the percentage of similarity or difference, using the operators “above of”, “equal to”, or “below of”.

According to [3], in the context of the Web, the unexpectedness existing in a piece of information is very expressive, and it considers that “a piece of information is *unexpected* if it is relevant but unknown to the user, or it contradicts the user's existing beliefs or expectations".

The used methods for the similarities discovery use the cosine measure algorithm between pages and sites contents. For unexpectedness, the formulas showed in "Equation 1" and in "Equation 2" are applied between terms of pages and pages of sites respectively.

$$UnexpT_{r,i,j} = \begin{cases} 1 - \dfrac{tf_{r,j}}{tf_{r,i}} & \text{if} \quad {tf_{r,j}}/{tf_{r,i}} \leq 1 \\ 0 & \text{if not} \end{cases} \tag{1}$$

$$UnexpP_i = \frac{\sum_{r=1}^{m} UnexpT_{r,c,u}}{m} \tag{2}$$

4.3 Classification

The Classification module serves to organize documents for topics in accordance with the subject treated for the document. The typical classification example is book organization in a library, where hierarchic catalogues are created. This process is slow and onerous, therefore it depends on a manual classification, where the responsible person reads the document and identifies the subject.

In this context, the use of automatic or half-automatic techniques of classification, which assists and speed the human work is necessary.

The classification process of organized texts in sites or pages does not use a methodology directed toward the most automatic classification possible. Two methods were considered in this work to provide the classifier, according to the pages and site organization. The first one is the junction of hierarchic and not-hierarchic document classifiers presented in [13–15], using a data structure based in a tree of categories, with few levels, preferably two, and the results of the searches are only classified in leaves of this tree, called *support vector machines* (SVM). The second one is the K-Nearest Neighbor (*k*-NN) that is a simple algorithm that stores all available data points (examples) and classifies new data points based on a similarity measure.

The classification process based on SVM is incremental and selective, guided for the hierarchic organization of the categories. In this kind of classification, particular characteristics of documents, such as tags and links are considered by the method. The documents are divided in two sets defined as training and tests bases. The training base is used for the classification algorithm to get the characteristics of the collection's categories. The test base validates the classifier performance, determining the categories the new document belongs. In the analysis phase, the algorithm performance is measured, in accordance with the obtained result with the original documents classification (manual classification). Although this method had been shown to be efficient and effective for classification, the researchers found little advantage in accuracy for hierarchical models over flat models. On other hand, others researchers have also shown that the use of SVM in context features (especially hyperlinks) can improve the classification performance significantly.

The k-Nearest Neighbor is one of the most popular algorithms for text categorization and belongs to the class of "lazy" algorithms. There is no process of learning a model. The examples are simply stored as the data is collected. The k-NN algorithm is suited for the regression problems as well, and achieves very good performance in the experiments on different data sets. However, the system performance is very sensitive to the choice of the parameter k.

Some variants for the *k*-NN algorithm are presented in [5, 16], whose profit is obtained through the choice of k value or in the terms of weight adjustment to be used in the calculation of Euclidean distance.

Considering all these characteristics presented for the methods, a *k*-NN algorithm was implemented, as it is better adjusted to the Web data set.

4.4 Clustering

The Clustering module executes the document grouping, in accordance with the similarities presented. The algorithm considers a document as being only one page or the whole site. The groups are created based in the characteristics of a documents set selected by the user (pages or sites).

Each document is associated with a characteristics vector or with dominant subjects or keywords lists and a relative importance measure of each subject or keyword, that describe it in a simplified way, with relation to its content. This simplified description is used for the algorithm, adding the document to the appropriated cluster [12, 17].

The algorithm presented in [18] is used for partitioning (or clustering) *N* data points into *K* disjoint subsets S_j containing N_j data points so as to minimize the sum-of-squares criterion, as showed in "Equation 3",

$$J = \sum \sum_{j=1}^{k} \left| x_n - \mu_j \right|^2 \qquad (3)$$

where x_n is a vector representing the n^{th} data point and μ_j is the geometric centroid of the data points in S_j. In general, the algorithm does not achieve a global minimum of *J* over the assignments. In fact, since the algorithm uses discrete assignment rather than a set of continuous parameters, the "minimum" it reaches cannot even be properly called a local minimum. Despite these limitations, the algorithm is used fairly frequently as a result of its easy implementation.

The algorithm consists of a simple re-estimation procedure as follows. First, the data points are assigned at random to the *K* sets. Then the centroid is computed for each set. These two steps are alternated until a stopping criterion is met, i.e., when there is no further change in the assignment of the data points.

A clustering method can be applied to documents, to group similar document representations, and is useful for the development of taxonomies used in search engines as Yahoo, or in search of similar documents to a document origin. In this work, this module is useful for comparing similar sites.

5 Case study

The case study is an *ongoing work* that serves to discover, between some computer science companies Web sites, the degree of experience of each company for one determined service, as well as identifying the set of customers for each company. Thus, it is possible to investigate the effected attendance of the customers, and also to identify the technical experience of each company.

For this, a list of activities was identified:

- Selection of companies to be investigated;
- Download of respective Web sites;
- Similarity identification between sites;
- Classification of Web sites contents;
- Results analysis.

The results still are preliminary; however, they show the efficiency of the tool in the solution of some questions applied to the knowledge search in a textual environment.

6 Conclusions

The presented work adds some functionality used in text mining, not found in a unique tool. The algorithms have shown good performance for the tested document collection, however it is not possible to generalize the results for great volumes of information.

It should be noted that some algorithms are in the test phase, not being able to be used even though a favorable results evaluation was obtained.

Although the experiments cannot be compared with others of the same category, the obtained results for some functionality are sufficiently interesting, stimulating the used methods refinement and the continuity of the research.

In general, this work can be considered relevant in the text mining area, since few developed works exist, related to Web knowledge discovery for Portuguese language.

Many improvement can be made in this environment, e.g., comparison between versions of the same site, identifying and showing the type of existing difference between them, local classification of a document in more than one category, and the use of a classification tree with many hierarchical levels, beyond other functionalities.

References

[1] Agrawal, R., Srikant, R., 1994. Fast Algorithms for Mining Association Rules, In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, Proceedings of 20th Int. Conf. Very Large Data Bases (VLDB), pages 487–499.

[2] Johansson, U., Niklasson, L., König, R., 2004. Accuracy vs. Comprehensibility in Data Mining Models. In *Proceedings of the Seventh International Conference on Information Fusion* (fusion2004), pp. 295-300.

[3] Liu, B., Ma, Y., Yu, P. S., 2001. Discovering Unexpected Information from Your Competitors' Web Sites. In *Proceedings of The Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001)*, pp 144-153.

[4] Orengo, V. M., Huyck, C., 2001. A Stemming Algorithm for the Portuguese Language. In *Proceedings. Eighth International Symposium on String Processing and Information Retrieval (SPIRE 2001)*, pp 186 – 193.

[5] Han, E. H., Karypis, G., Kumar, V., 2001. Text Categorization Using Weight Adjusted *k*-Nearest Neighbor Classification, in Proceedings of 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), pp.53-65.

[6] Weiss, S., Indurkhya, N., Zhang, T., Damerau, F. J., 2005, *Text Mining – Predictive Methods for Analyzing Unstructured Information*, Springer.

[7] Campione, M., Walrath, K., 1996. *The Java Tutorial: Object-Oriented Programming for the Internet*. SunSoft Press.

[8] http://java.sun.com/.

[9] http://java.sun.com/products/jsp/.

[10] Chang, J., Healey, M. J., McHugh, J. A. M., Wang, J. T. L., 2001. *Mining the World Wide Web An Information Search Approach*. Kluwer Academic Publishers.

[11] Porter, M.F., 1980. An Algorithm for Suffix Stripping. *Program***,** vol.14, n. 3, pp. 130-137.

[12] Lopes, Maria Célia S., 2004. *Mineração de Dados Textuais Utilizando Técnicas de Clustering para o Idioma Português*, Tese de Doutorado, COPPE/UFRJ.

[13] Dumais, S., Chen, H., 2000. Hierarchical Classification of Web Content. In Proceedings of SIGIR-00*, ACM International Conference on Research and Development in Information Retrieval.*

[14] Sum, A., Lim, E., Ng, E., 2002. Web Classification using Support Vector Machine. In Proceedings of WIDM'2002, *ACM Fourth International Workshop on Web Information and Data Management*, pp. 96-99.

[15] Zhang, D., Lee, W. S., 2004. Web Taxonomy Integration using Support Vector Machines. In *Proceedings of the Thirteenth International World Wide Web Conference*, pp 472-481.

[16] Baoli, L., Shiwen, Y., Qin, L., 2003. An Improved *k*-Nearest Neighbor Algorithm for Text Categorization, In *Proceedings of the 20th International Conference on Computer Processing of Oriental Languages, Shenyang, China.*

[17] Joachims, T., 1998. Text Retrieval with support vector machine: learning with many relevant features. In Proceedings of ECML-98*, 10th European Conference on Machine Learning*, pp. 137-142.

[18] Bishop, C. M., 1995. *Neural Networks for Pattern Recognition*. Oxford, England: Oxford University Press.