Using Stemming Algorithms on a Grid Environment

Valeriana G. Roncero, Myrian C. A. Costa, and Nelson F. F. Ebecken

COPPE/Federal University of Rio de Janeiro
P.O. Box 68516, 21945-970, Rio de Janeiro, RJ, Brazil.
{valery,myrian}@nacad.ufrj.br, nelson@ntt.ufrj.br

Abstract. Stemming algorithms are commonly used in Information Retrieval with the goal of reducing the number of the words which are in the same morphological variant in a common representation. Stemming analysis is one of the tasks of the pre-processing phase on text mining that consumes a lot of time. This study proposes a model of distributed stemming analysis on a grid environment to reduce the stemming processing time; this speeds up the text preparation. This model can be integrated into grid-based text mining tool, helping to improve the overall performance of the text mining process.

Key words: grid environment, distributed computing, text mining, stemming analysis

1 Introduction

The enormous amount of information stored in unstructured texts cannot simply be used for further processing by computers, which typically handle text as simple sequences of character strings. Text mining is the process of extracting interesting information and knowledge from unstructured text. It runs several processes, such as document collection, pre-processing and preparation, pattern discovery and evaluation and interpretation of the results.

Text mining techniques gained considerable importance as a technology to retrieve data from huge amount of digitally stored documents [1]. In order to extract useful patterns [2] pre-processing tasks and algorithms are required. Stemming analysis, which is used in this paper, is one of the pre-processing tasks of text mining.

Due to the large quantity of documents, pre-processing tasks are computationally intensive. In order to reduce the time spent in pre-processing, in particular stemming analysis, we distribute the stemming analysis on a grid environment. This environment is a geographically distributed computation infrastructure composed of a set of heterogeneous resources.

2 Text Mining

Text mining is a relatively new practice derived from Information Retrieval (IR) [3, 4] and Natural Language Processing (NLP) [5]. The strict definition of text mining includes only the methods capable of discovering new information that is not obvious

or easy to find out in a document collection, i.e., reports, historical documents, e-mails, spreadsheets, papers and others.

Text mining executes several processes, each one consisting of multiple phases, which transform or organize an amount of documents in a systematized structure. These phases enable the use of processed documents later, in an efficient and intelligent manner. The processes that compose the text mining can be visualized in Figure 1 that is summarized version of the figure model from [6] on page 6.

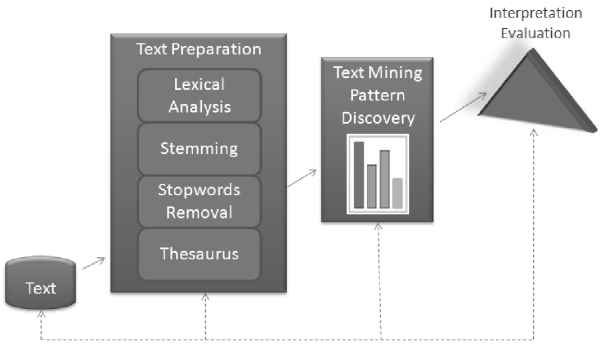

Fig. 1. Summary of the text mining phases

The text mining processes are divided in the following phases:

1. Document collection: consists of the definition of the set of the documents from which knowledge must be extracted.
2. Pre-processing and preparation: consists of a set of actions that transform the set of documents in natural language into a list of useful terms. Then from these terms will be identified and selected the relevant terms.
3. Text Mining Pattern Discovery: consists of the application of machine learning techniques to identify patterns that can classify or cluster the documents in the collection.
4. Evaluation and interpretation of the results: consists of the analysis of the results.

The pre-processing phase in text mining is essential and usually time consuming. As texts are originally non-structured series of steps are required to represent them in a format compatible with knowledge extraction methods and tools.

2.1 The Stemming Process

The stemming process is an important pre-processing task before indexing input documents for text mining. The term stemming refers to the reduction of words to their roots, so that, for example, different grammatical forms or declinations of verbs are identified and indexed (counted) as the same word. For example, stemming will ensure that both takes and take will be recognized by the program as the same word [7]. In

most cases, morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of Information Retrieval applications. For this reason, a number of so-called stemming Algorithms, or stemmers, have been developed, which attempt to reduce a word to its root form.

Lovins [8] described the first stemmer developed specifically for Information Retrieval applications and introduced the idea of stemming based on a dictionary of common suffixes. This algorithm stimulated the development of many subsequent algorithms [9, 10] and, more generally, the use of stemming as a general tool in the Information Retrieval area [10–14].

In this model two well-known stemmer algorithms will be implemented: Porter stemmer and Paice/Husk stemmer. The Porter stemming algorithm [15] is a process for removing the commoner morphological and suffixes from words. Its main use is as part of a term normalization process that is usually done when setting up Information Retrieval systems [16]. The Paice/Husk stemmer [17] is iterative and uses a single table of rules, each rule may specify the removal or replacement of an ending. The rules are grouped into sections corresponding to the final letter of the suffix; this means that the rule table is accessed quickly by looking up the final letter of the current word or truncated word.

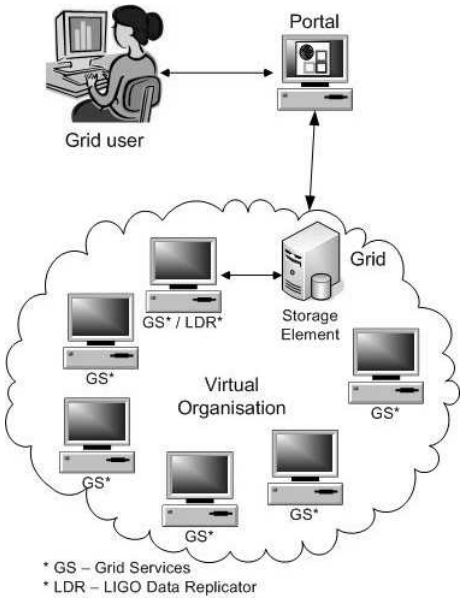

Fig. 2. The Grid NACAD infrastructure

3 Grid Environment

A grid is a geographically distributed computation infrastructure composed of a set of heterogeneous machines, often with separate policies for security and resource use

[18], that users can access via a single interface. Grids therefore, provide a common resource-access technology and operational services across widely distributed virtual organizations composed of institutions or individuals that share resources. Today grids can be used as effective infrastructures for distributed high-performance computing and data processing [19]. Figure 2 shows the Grid NACAD infrastructure that will be used in this work.

In this work we use the Globus Toolkit 4 (GT4) [20], which is a widely used middleware in scientific and data-intensive grid applications, and is becoming standard for implementing grid systems. The toolkit addresses security, information discovery, resource and data management, communication, fault-detection, and portability issues. Today, Globus and the other grid tools are used in many projects worldwide. Although most of these projects are in scientific and technical computing fields, and a growing number of grid projects in education, industry, and commerce are being implemented.

4 Distributed Stemming Analysis on a Grid

The pre-processing phase, in particular the stemming task, is very time consuming. This is so because of the large number of words that a document collection contains. To reduce the time spent for stemming, we distribute the documents on a grid environment to process the stemming simultaneously in the grid nodes.

The Globus Toolkit provides a number of components for performing data management. Data management tools (GridFTP, RFT, RLS) are concerned with the location, transfer, and management of distributed data [21]. GridFTP protocol provides a secure way to transfer data in a grid. RFT (Reliable File Transfer) is a Web Services Resource Framework (WSRF) [22] compliant web service for managing multiple data transfers. The Replica Location Service (RLS) [23] maintains and provides access to mapping information from logical names for data items onto target names. These target names may represent physical locations of data items, or an entry in the RLS may map to another level of logical naming for the data item. The RLS is intended to be one of a set of services for providing data replication management in grids. In additional of this components, the LIGO Data Replicator (LDR) [24, 25] will be used. LDR is a collection of some components provided by the Globus project with some extra logic to pull the components together, this minimum collection of components is necessary for fast, efficient, robust, and secure replication of data. The Globus components include are: GridFTP, Globus Replica Location Service (RLS) and a metadata service developed by the LDR team but based on a prototype Globus Metadata Catalog Service (MCS) [26] for organizing useful information about the data files, especially as it pertains to when and where the data should be replicated.

Figure 3 shows the distributed stemming analysis on a grid model. The grid user owns a grid certificate, which provides him with the grid credentials [27] to log into the grid and submit jobs to it, which is done by means of a Portal, accessible from the user's workstation. After logged in, the user can access his documents or public documents that are stored in the grid. He submits to the Portal information about the documents that will be analyzed (1). The Portal uses the LDR queries to find out whether there is a local copy of the documents, if not, RLS tells to the Portal where the documents

are in the grid (2). Then the LDR system generates a request to copy the documents to the local storage system and registers the new copy in the local RLS server. The grid nodes receive from the Portal the phases to run the stemming task (3) and using the RFT service it copies the replicas of the documents from the storage to the grid nodes (4). When the stemming task is concluded, the Portal collects all sets of documents from each node and returns the result of the stemming to the user, who stores the documents in his grid account area.

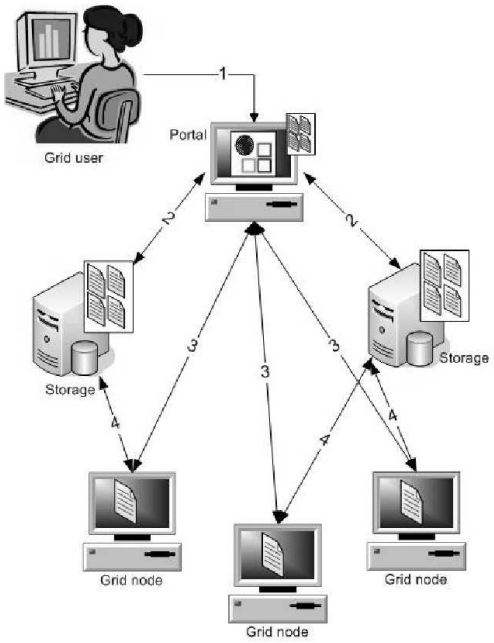

Fig. 3. Distributed stemming analysis model

5 Summary

In this paper we presented a model of a distributed stemming analysis. This model focuses on reducing the stemming task processing time, using a grid environment to distribute the documents to speed up the stemming task within a group of documents. The next step is to develop this model and integrate it to a text mining system through a grid service using the Globus Toolkit middleware in the Grid NACAD.

Acknowledgments. The authors would like to thank the High Performance Computing Center (NACAD) at the Graduate School and Research in Engineering (COPPE), Federal University of Rio de Janeiro for providing the computational resources for this research and The National Research Council of Brazil (CNPq) for financial support.

References

1. Hearst, M.A.: Untangling text data mining. In: Proceedings of the 37^{th} Annual Meeting on Computational Linguistics, Association for Computational Linguistics (1999) 3–10
2. Konchady, M.: Text Mining Application Programming. Charles River Media (2006)
3. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill Book Company (1983)
4. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press Books (1999)
5. Kao, A., Poteet, S.R.: Natural Language Processing and Text Mining. Springer-Verlag (2007)
6. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann (2001)
7. Manning, C.D., Schuetze, H.: Foundations of Statistical Natural Language Processing. MIT Press (1999)
8. Lovins, J.B.: Development of a stemming algorithm. Mechanical Translation and Computational Linguistics **11**(1/2) (1968) 22–31
9. Lennon, M., Peirce, D.S., Tarry, B.D.: Development of a stemming algorithm. Mechanical Translation and Computational Linguistics **3**(4) (1981) 177–183
10. Porter, M.F., Tait, J.I.: Charting a new course: Natural language processing and information retrieval. In: Essays in Honour of Karen Spärck Jones. Springer-Verlag (2005) 39–68
11. Frakes, W.B., Fox, C.J.: Strength and similarity of affix removal stemming algorithms. In: ACM SIGIR Forum. Volume 37. (2003) 26–30
12. Harman, D.: How effective is suffixing? Journal of the American Society for Information Science **42**(1) (1991) 7–15
13. Hull, D.A.: Stemming algorithms: a case study for detailed evaluation. Journal of the American Society for Information Science **47**(1) (1996) 70–84
14. Krovetz, B.: Viewing morphology as an inference process. Artificial Intelligence **118**(1/2) (2000) 277–294
15. Porter, M.F.: An algorithm for suffix stripping. Program (July 1980)
16. Porter, M.F.: The porter stemming algorithm http://tartarus.org/∼martin/PorterStemmer/index.html.
17. Paice, C.D.: Another stemmer. SIGIR Forum **24**(3) (1990) 56–61
18. Qi, L., Jin, H., Foster, I., Gawor, J.: Hand: Highly available dynamic deployment infrastructure for globus toolkit 4. http://www.globus.org/ alliance/∼publications/papers/HAND-Submitted.pdf
19. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the grid: Enabling scalable virtual organizations. Intl. J. Supercomputer Applications **15**(3) (2001)
20. The Globus Toolkit. http://www.globus.org/toolkit/
21. GT4 Data Management. http://www.globus.org/toolkit/docs/4.0/data/
22. The WS-Resource Framework. http://www.globus.org/wsrf/
23. Replica Location Service. http://www.globus.org/toolkit/data/rls/
24. LIGO Scientific Collaboration Research Group: Ligo data replicator. http://www.lsc-group.phys.uwm.edu/LDR/
25. Chervenak, A., Schuler, R., Kesselman, C., Koranda, S., Moe, B.: Wide area data replication for scientific collaborations. In: Proceedings of 6^{th} IEEE/ACM International Workshop on Grid Computing (Grid2005). (November 2005)
26. Metadata Catalog Service. http://www.globus.org/grid_software/data/mcs.php
27. GT 4.0: Security: Pre-Web Services Authentication and Authorization. http://www.globus.org/toolkit/docs/4.0/security/prewsaa/