Biomedical Text Mining Using a Grid Computing Approach

Marcello Castellano[1,2], Giuseppe Mastronardi[1,2], Giacinto Decataldo[1], Luca Pisciotta[1], Gianfranco Tarricone[1], Lucia Cariello[1,2], and Vitoantonio Bevilacqua[1,2]

[1] Dipartimento di Elettrotecnica ed Elettronica Politecnico di Barivia Orabona, 4 70125- Bari-Italy
castellano@poliba.it
[2] e.B.I.S. s.r.l. (electronic Business in Security), Spin-Off of Polytechnic of Bari, Str. Prov. per Casamassima Km. 3-70010 Valenzano (BA)-Italy

Abstract. Extracting useful information from a very large amount of biomedical texts is an important and difficult activity in biomedicine field. Data to be examined are generally unstructured and the available computational resources do not still provide adequate mechanisms for retrieving and analyse very large amount of contents. In this paper we present a rule-based system for Text Mining process applied in biomedical textual documents. This application requires a strongly use of the computational resource to perform intensive operations. We propose a grid computing approach to improve application performance.

Keywords: Text Mining; Computational Grid; SIMD; Knowledge Discovery; Biomedical Document Analysis.

1 Introduction

The problem of discovering useful knowledge from unstructured text, is attracting increasing attention. The process of extracting interesting and not-retrieval patterns or knowledge from unstructured text documents is known as Knowledge Discovery in Text (KDT). One of the most difficult applications of Knowledge Discovery in Texts is Text Mining (TM) of biomedical papers: the sheer volume of biomedical research output makes TM a necessity, while the importance of this research requires extremely high retrieval precision. TM examines the relationships between specific kinds of information contained both within and between documents. TM concentrates on solving a specific problem in a specific domain identified a priori. TM can aid database curators by selecting articles most likely to contain information of interest, or potential new treatments. The goal of biomedical text mining is therefore to allow researchers to identify needed information more efficiently, uncover relationships obscured by the sheer volume of available information, and in general shift the burden of information overload from the researcher to the computer by applying algorithmic, statistical and data management methods to the vast amount of biomedical knowledge [1,2,3,4].

D.-S. Huang et al. (Eds.): ICIC 2008, LNAI 5227, pp. 1077–1084, 2008.

In this paper we discuss a Text Mining Process applied in biomedical documents analysis to recognize biological entities based on rule system. Moreover we discuss a grid computing approach to the problem in order to improve the TM application performance. Finally, experimental results and conclusions are presented.

2 Text Mining for Biomedical Document Analysis

Significant progress has been made in applying text mining to the following text analysis: named entity recognition, text classification, terminology extraction, relationship extraction and hypothesis generation.

The task of named entity recognition appears straightforward. The goal is to identify, within a collection of text, all of the instances of a name for a specific type of thing: for example, all of the drug names within a collection of journal articles, or all of the gene names and symbols within a collection of abstracts. The idea is that recognising biological entities in text allows for further extraction of relationships and other information by identifying the key concepts of interest and allowing those concepts to be represented in some consistent, normalised form. This task has been challenging for several reasons. First, there does not exist a complete dictionary for most types of biological named entities, so simple text matching algorithms do not suffice. In addition, the same word or phrase can refer to a different thing depending upon context. Biological entities may also have multi-word names, like carotid artery, so the problem is additionally complicated by the need to determine name boundaries and resolve overlap of candidate names.

Text classification attempts to automatically determine whether a document or part of a document has particular characteristics of interest, usually based on whether the document discusses a given topic or contains a certain type of information. Typically the information of interest is not specified explicitly by the users and, instead, they provide a set of documents that have been found to contain the characteristics of interest (the positive training set), and another set that does not (the negative training set). Text classification systems must automatically extract the features that help determine positives from negatives and apply those features to candidate documents using some kind of decision-making process.

Paralleling the growth of the increase in biomedical literature is the growth in biomedical terminology. Because many biomedical entities have multiple names and abbreviations, it would be advantageous to have an automated means to collect these synonyms and abbreviations to aid users doing literature searches. Furthermore, other text-mining tasks could be done more efficiently if all of the synonyms and abbreviations for an entity could be mapped to a single term representing the concept. Most of the work in this type of extraction has focused on uncovering gene name synonyms and biomedical term abbreviations.

The goal of relationship extraction is to detect occurrences of a pre-specified type of relationship between a pair of entities of given types. While the type of the entities is usually very specific, like genes, proteins or drugs, the type of relationship may be very general, like any biochemical association, or very specific, like a regulatory relationship.

While relationship extraction focuses on the extraction of relationships between entities explicitly found in the text, hypothesis generation attempts to uncover relationships that are not present in the text but instead are inferred by the presence of other more explicit relationships. The goal is to uncover previously unrecognised relationships worthy of further investigation [5,6].

Fig. 1 shows Knowledge Discovery in Text process steps. Knowledge Discovery in Text can be visualized as consisting of two phases: Text Refining that transforms free-form text document in a chosen Intermediate Form and Text Mining that deduces patterns or knowledge from the Intermediate Form [2,7,8,9,10].

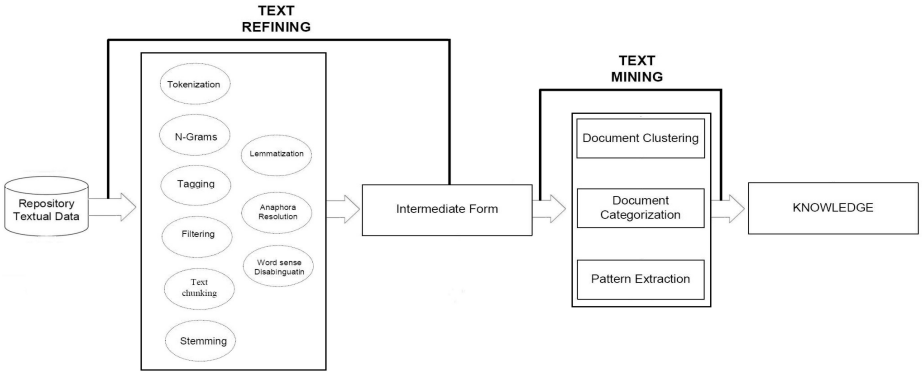

Fig. 1. Knowledge Discovery in Text

2.1 Building the Association Rule Induction

Association discovery is a central activity of the text mining phase. It is the identification process of meaningful correlations among frequent whole data and can being applied to textual structured form. A rule consists of a left-hand side proposition, antecedent, and a right-hand side, consequent. Both sides consist of Boolean statements or predicates. The rule states that if the left-hand side is true, then the right-hand is also true. A probabilistic rule modifies this definition so that the right-hand side is true with probability p, given that the left-hand side is true. An association rule is described by the form:

$$X \Rightarrow Y$$

where X and Y are predicates or set of items. As the number of produced associations might be huge, and not all the discovered associations are meaningful, two probability measures, called support and confidence, are introduced to discard the less frequent associations in the database.

The support is defined as:

$$fr(X \Rightarrow Y) \tag{1}$$

while the confidence is:

$$c(X \Rightarrow Y) = \frac{fr(X \wedge Y)}{fr(X)} \tag{2}$$

The accuracy of the association rule, when regarded in terms of conditional probability, can be seen as a maximum likelihood (frequency-based) estimate of the conditional probability that Y is true, given that X is true.

Through association rules it happens the classification or text categorization of the analyzed text. The definition of accurate rules allows to the TM tool to analyze the whole text and to decide if the tokens, in which the text is divided, belongs to a class, lemma, or another. Any TM rule founds its principle on the matching, that is to say, "keywords lists" are created each of which contains a subset of words, specifications and not, of the universe, in our case of the Biomedicine. The matching is not enough when one of these words, token, is present in more lists or there isn't in some lists, in such case it needs to understand what is the correct meant to give to the token and therefore the TM rules must be define so that to analyze the tokens that follow or precede the token to classify. The matching rule is the rule that seeks the presence in a keywords lists for every considered token and associates, to the token, the name of the same list:

IF *token belongs to a list* AND *token it doesn't belong to other lists*
THEN *the meaning to the token is the name of the same list*

The problem is when the token is present in more or no lists, in such case grammatical rules and interpretation rules are necessary. For example we consider the word "Fanconi" this can be in two different contexts, in the sentence "…Nicola Fanconi restaurant…. " or "…discovery of new drugs for the anaemia of Fanconi…. ", to this point the simple matching rule is not enough to give an unequivocal meaning to the token "Fanconi" being in two lists "Surname" and "Pathology". To give a meaning to this token is needed categorization the lemmas, that is, understanding the meaning of the following and the precedent token, an example is the following rule:

IF *token belongs to the list "surname"* AND *token belongs to the list "pathology"*
IF *preceding token* OR *following token belongs to the list "names"*
THEN *token preceding/following & token means "first name"*
ELSE *token means "pathology'*

An example of rule is in which a token is not in any list, in such case it is need to understand what meant it can have, for example, if we consider "Anatomy and Orthopaedics Anderson-Fabry though CT" the token "Anderson-Fabry" could not be present in any list. To this point it is possible to understand which meant to give to this token, in our case, being followed by a key CT (compression therapy), the token is the name of one syndrome that brings the name of the physicians that have discovered her; then a rule of such case is:

IF *token is not present in any list*
IF *preceding token belongs to the list "therapy/drugs"*
THEN *token & token preceding/following mean "pathology"*

Another rule is:

IF *token belongs to the list 'human anatomy'* AND *preceding token belongs to the list "therapy/drugs"*
THEN *token & token preceding/following mean "symptom"*

3 Grid Computing in Text Mining

In Figure 2a is shown a schema of the typical Job produced by a TM tool. This is a single-instruction and multiple-data stream Job that reveals an intensive use of the CPU resource. In Figure 2b is shown the Job model here adopted to overcome the single CPU bound. It distributes several computing nodes the user commands and a slice of the whole data set. At the completion time each node came back the results.

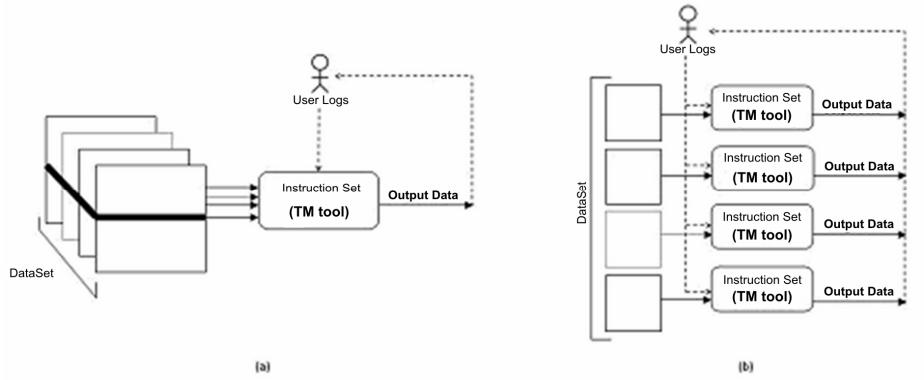

Fig. 2. Text Mining Job Models

In Figure 3 is shown the layered adapter system architecture designed as a software application between a grid middleware infrastructure and the user application to implement the Job model for Grid Computing.

In the hierarchical architecture, there is, to lowest level, the grid middleware with its services, and to next level the shell scripts that interoperate to grid middleware using services.

The functional management system layer is composed by:

- Grid node search system;
- A Load Balancer;
- User program modules ;
- User's module management system;
- A Transfer Optimizer;

The *node search system* effects the search of nodes which are available for execution.

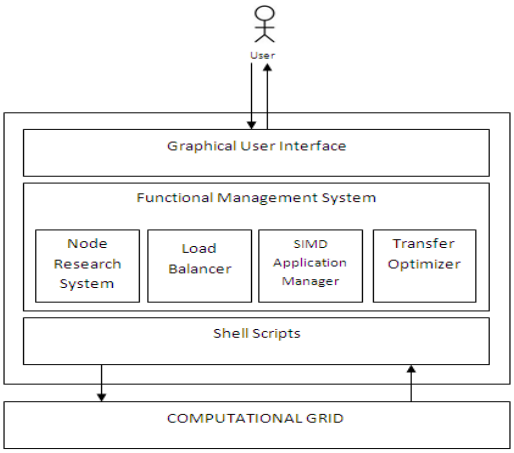

Fig. 3. The Adapter Hierarchical Architecture

The *Load Balancing system* analyzes the whole input data set and the computing nodes previously selected to split the work for parallel execution.

The *user program modules* are developed or available for the end user. They are programs which are distributed on the nodes together data for the execution. It receives and produce a file or a set text file.

The *program modules manager* allows to the user to add and to manage all modules that user wants to use. The manager produce its activity up on a set up of a configuration file. For example to taken into account the applicative modules for the symptoms and pathologies extraction a such configuration file must be updated.

The *transfer optimizer module* performs the data compression optimizing the communication time.

Upper layer, there is a graphical user interface, that supplies the direct and intuitive access to the system functionalities. Through simple click and, in completely transparent way to the user, the interface dialogues with the underlying modules. The communication with lower layer modules, is realized by a shell script passing the actual value of the parameter.

The software system architecture here presented is an effective solution to improve the performance of TM application in bioinformatics.

4 Experimental Results

The process of Text Mining starts from a set of 1000 scientific full text publications available on MedLine / Pubmed in pdf format. The process consists of three phases: text extraction from pdf documents, text mining rules application as described in section 2.1, results storage in a repository composing of identified biological entities (symptoms and pathologies). Text mining rules application is realized using GATE 4.0 tool JAVA APIs [13]. Figure 4 shows a sample of obtained text mining results. Results show the efficiency of the rules application.

Grid execution of text mining operation has been realized through a prototype development written in Java following the system architecture proposed in paragraph 3, using Globus Toolkit 4 grid middleware [14].

The use of standard Java APIs allows porting code on all computational grid nodes used for this experiment, provided that the experimental grid is here built using Globus Toolkit 4 on Gnu / Linux systems.

System efficiency has been evaluated through execution time estimated on a serial computer and then on a computational grid. Execution time has been estimated for a grid with a variable nodes number considering an input data set composed of 1000 biomedical documents with appropriate dimensions. Estimation of speedup indicator has been obtained. Speedup indicator is defined as the ratio between the mining process execution time on traditional serial machine, and on a computational grid. Figure 5 shows architecture speedup depending on grid node number.

Biomedical Document	Identified Pathology
The prevalence of tuberculosis in the state of acre	tuberculosis
The prevalence of tuberculosis in the state of acre	death
The prevalence of tuberculosis in the state of acre	aids
Potentially infectious residues at hemotherapy services [...]	hepatitis
The meaning of cancer in the everyday of women [...]	cancer

Biomedical Document	Identified Symptom
Haemophilus influenzae antibiotic resistant strains [...]	sinusitis
Haemophilus influenzae antibiotic resistant strains [...]	bronchitis
Haemophilus influenzae antibiotic resistant strains [...]	pneumonia
Efficacy of cow's kumiss in the treatment of large [...]	regression
Association between particulate air pollution and first [...]	regression

Fig. 4. Text Mining Results

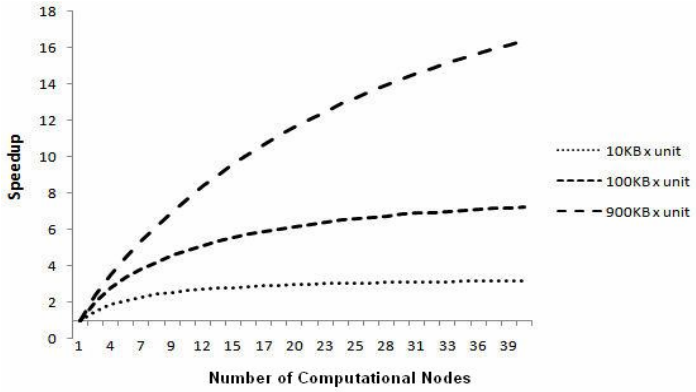

Fig. 5. Speedup depending on grid node number, considering a data set of 1000 documents from 10KB to 900KB each

5 Conclusions

In this paper has been presented a biomedical Text Mining using a grid computing approach. The TM application requires a strongly use of the computational resource and we proposed a grid based approach to improve the application performance.

References

1. Polajnar, T.: Survey of Text Mining of Biomedical Corpora (2006)
2. Tan, A.H.: Text Mining: The State of the Art and the Challenges. In: Zhong, N., Zhou, L. (eds.) PAKDD 1999. LNCS (LNAI), vol. 1574. Springer, Heidelberg (1999)
3. Prather, J.C., Lobach, D.F., Goodwin, L.C., Hales, J.W.: Medical data mining: knowledge discovery in a clinical data warehouse. In: Proc. AMIA Annu. Fall Symp, Division of Medical Informatics, Duke University Medical Center, Durham, North Carolina, USA, pp. 101–105 (1997)
4. Cohen, A.M., Hersh, W.R.: A Survey of Current Work in Biomedical Text Mining. Briefing in Bioinformatics 6 (2005)
5. Polanski, A., Kimmel, M.: Bioinformatics. Springer, Heidelberg (2007)
6. Hersh, W.: Evaluation of biomedical text-mining systems. Briefings in Bioinformatics 6(4), 344–356 (2005)
7. Manning, C.D., Schutze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
8. Ahonen, H.: Finding All Maximal Frequent Sequences in Text. In: ICML 1999 Workshop on Machine Learning in Text Data Analysis, Bled, Slovenia (1999)
9. Hotho, A., Numberger, A., Paab, G.: A Brief Survey of Text Mining. LDV Forum-GLDV Journal for Computational Linguistics and Language Technology 20(1), 19–62 (2005)
10. Mobasher, B., Cooley, R., Srivastava, J.: Creating Adaptive Web Sites Through Usage-Based Clustering of URLs (1999). In: Proc. of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX 1999) (1999)
11. Foster, I., Kesselmann, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan-Kaufmann edition (1998)
12. Castellano, M., Aprile, A., Mastronardi, G., Piscitelli, G., Dicensi, V., Giuseppe, D.G.: Simulating a Computational Grid. GESTS, International Transaction on Communication and Signal Processing (2007)
13. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002), Philadelphia (2002)
14. The Globus Alliance: Globus Toolkit 4, `http://www.globus.org/toolkit`