Grid-enabled Support for Classification and Clustering of Textual Documents

Martin Sarnovský, Peter Butka

Department of Cybernetics and Artificial Intelligence, Technical University of Košice, Letná 9, 042 00 Košice, Slovakia
Martin.Sarnovsky@tuke.sk, Peter.Butka@tuke.sk

Abstract: This paper presents the fusion of two approaches – Grid and Grid computing and text mining. GridMiner is a system developed at University of Vienna, and it is a framework for knowledge discovery process in the distributed Grid environment. JBOWL is a framework for text mining and information retrieval being developed at Technical University in Košice. Text mining provides some methods (including the classification and clustering) in order to automatically extract the relevant knowledge contained in the textual documents. The inclusion of the Grid text mining services into the Grid-based knowledge discovery system can support problem solving process on such a system. Motivation of the presented work is to use the Grid computational capabilities for text mining tasks and text classification or clustering in particular.

Keywords: text-mining, grid services, text clustering, text classification, text categorization

1 Introduction

Motivation of this work is to use the Grid computational capabilities for text mining tasks. Some of the methods are time-consuming and use of the Grid infrastructure can bring significant benefits. Implementation of text mining techniques in distributed environment allows us to access different geographically distributed data collections and perform text mining tasks in parallel and distributed fashion.

The main effort of this article is to give in theoretic description process of processing textual documents and their classification or clustering, to describe progress of their processing in grid environment, their testing, evaluation of experimental results and possible manners of availing for improvement classification and clustering results.

2 Data Mining in Grid Environment

The one of the very first projects in the area of Grid-based knowledge discovery is *TeraGrid.* The TeraGrid is such a virtual supercomputer, built from four individual clusters (San Diego Supercomputing Center, National Center for Supercomputing Applications, Caltech a Argonne National Lab) that allows to create the Grid architecture. It offers the access to very large amount of the data, and one of the pilot applications in this project is knowledge discovery in these large data sets.

One of the various projects is the *ADaM*[1] (Algorithm Development and Mining) project, which is agent-based platform for knowledge discovery researched at Alabama University. Main goal of this project was to process the hydrological databases in parallel fashion from different resources.

Very similar is the *Discovery Net*[2] project, aimed at possibilities of connection of different already existing systems for knowledge discovery with Grid infrastructure based on Globus Toolkit. Various activities are involved in this project, including the activities of building the Grid infrastructure for data mining, and also text mining in Grid environment. Main goal is to use such a system for mining in medical databases.

NaCTeM[3] *(*The National Centre for Text Mining) is the first publicly-funded text mining centre in the world. They provide text mining services in response to the requirements of the UK academic community. The goal of this project would be to investigate needs and to develop an infrastructure that will enable various text mining applications to work in the GRID environment. This topic would include investigation into the roles of text and text mining for the Semantic Web and the Semantic GRID, and vice versa.

3 Grid and Grid Computing

Grid computing represents the natural evolution of distributed computing and parallel-processing technologies. Grid is a technology, that allows from geographically distributed computational and memory resources create an ***universal computing system*** with extreme performance and capacity. System has the features of global worldwide computer, where all of the components are connected together via Internet. For the user it appears like a common workstation, but some segments of a solved task are computed in different parts of

[1] http://www.sciencetools.com/
[2] http://www.discovery-on-the.net/
[3] http://www.cse.salford.ac.uk/nactem/

system [1]. The main advantage of the Grid is high efficiency of using technological capacity.

The benefit of the Grid is high effectiveness of using associated technological capacities of creative users potential, the safety, the reliability, the effectiveness and high level of transportability for computational applications.

The main building block of the Grid is network. Geographically distributed resources are linked together via networks. Networks also allow them to be used collectively. Networks connect the resources on the Grid, the most prevalent of which are computers with data storage. Computational elements can be on any level of power and capability. Some of Grids involve nodes that are high-performance machines or clusters. These Grid nodes provide major resources for simulation, analysis, data mining, text mining and other activities.

4 Text-Mining Tasks

4.1 Classification of Textual Documents

Text categorization is a process of automatic assigning of textual documents into categories based on their meaning. One of the ways how to improve the classification is using information that provides more detailed view on the documents.

Text classification is the problem of assigning a text document into one or more topic categories or classes based on document's content. Traditional approaches to classification problems usually consider only the uni-label classification problem. It means that each document in collection has associated one unique class label. We face the problem of assigning the document into more than one single category. One sample can be labeled with a set of classes, so techniques for the multi-label problem have to be explored. Especially in text mining tasks, it is likely that data belongs to multiple classes. Most frequently used approach to deal with multi-label classification problem is to treat each category as a separate binary classification problem, which involves learning a number of different binary classifiers and use an output of these binary classifiers to determine the labels of a new example.

In the work reported in this paper, we used the decision trees algorithm based on the Quinlan's C4.5. A decision tree classifier is a tree with internal nodes labeled by attributes (words), branches departing from them are labeled by tests the weight that attribute has in the document, and leafs represent the categories. Decision tree classifies the unknown example by recursively testing of weights in the internal nodes, until a leaf is reached.

4.2 Clustering of Textual Documents

Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. From a machine learning perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning, and the resulting system represents a data concept. The goal of text clustering is to find some unseen categories (or clusters) in the set of analysed documents. In the work presented in this paper we used two clustering algorithms – k-means [7] and GHSOM (Growing Hierarchical Self-Organizing Map) [8].

The k-means algorithm is by far the most popular clustering tool used in scientific and industrial applications. The name comes from representing each of k clusters C_j by the mean (or weighted average) c_j of its points, the so-called centroid. While this obviously does not work well with categorical attributes, it has the good geometric and statistical sense for numerical attributes. The sum of discrepancies between a point and its centroid expressed through appropriate distance is used as the objective function. We have used classic version of k-means iterative optimization reassigns points based on more detailed analysis of effects on the objective function caused by moving a point from its current cluster to a potentially new one. If a move has a positive effect, the point is relocated and the two centroids are recomputed.

One of the ways for automatically organizing a set of documents on a map display is provided by the self-organizing map (SOM), a popular unsupervised neural network. The self-organizing map is an artificial neural network model that is well suited for mapping high-dimensional data into a 2-dimensional representation space. Fundamental feature of SOM (and related architectures) is their topology preserve mapping. The basic problem of SOM is that they have fixed architecture. The model being closest to the SOM is the *Growing Grid*, introduced by Fritzke, where a SOM-like neural network grows dynamically during training. Another possibility is to use a hierarchical structure of independent SOM, where for every unit of a map a SOM is added to the next layer. This means architecture called *Hierarchical Feature Map*. The *Growing Hierarchical Self-Organizing Map* (GHSOM) combines the benefits of this neural network models. It means hierarchical architecture where each layer is composed of independent SOM that adjust their size according to the requirements of the input data.

5 Used Tools

5.1 JBOWL

JBOWL (Java Bag-Of-Words Library)[4] is a software system developed in Java for support of information retrieval and text mining [2]. The system is being developed as open source with the intention to provide an easy extensible, modular framework for pre-processing, indexing and further exploration of large text collections, as well as for creation and evaluation of supervised and unsupervised text-mining models. JBOWL supports the document preprocessing, building the text mining model and evaluation of the model.

Architecture of JBOWL has three base components that may be implemented as one executable or in a distributed environment ([2], [3], [5]):

- *application programming interface (API),*
- *text mining engine (TME),*
- *mining object repository (MOR).*

5.2 GRIDMINER

GRIDMINER[5] is a service based software architecture for distributed data mining in the Grid environment. The aim of GridMiner is to provide a system, which supports the single steps of the knowledge discovery process by a set of OGSA services covering data integration, data preprocessing, data mining methods and visualization of results and integrate them to a novel service-oriented grid-aware application. This architecture is being developed on top of the Globus 3.0 toolkit, which provides a fundamental platform for grid services.

The aim of GridMiner project is to give to an user a tool of knowledge discovery process in the distributed Grid environment. The system provides also a graphical user interface that hides the complexity of the Grid, but still offers the possibility to interfere during the execution, control the task and visualize the results.

The architecture of GridMiner consists of three layers [6]:

- *the Grid layer,*
- *the Web layer and*
- *the User environment.*

[4] http://sourceforge.net/project/showfiles.php?group_id=147868/
[5] http://www.cse.salford.ac.uk/nactem/

The data mining process is supported by several services able to perform data mining tasks. Data mining services includes sequential, parallel and distributed implementation of data mining algorithms.

6 Implementation

We have used Main task was to enhance the existing GridMiner system with the text mining support. GridMiner is a service-based system, so I decided to integrate the text mining support as a Grid service using JBOWL library. Since JBOWL already has implemented the tools for document preprocessing, classification and evaluation, my main goal was to design a service interface and simple client for that service.

The first goal was to design a sequential version of Text Classification Service based on the multi-label algorithm implemented in the JBOWL library, which is discussed below. While the preprocessing phase is implemented as a pure sequential method, the classification model can build the final model parallelizing the partial binary classifiers. Other methods were also implemented to provide term reduction and model evaluation.

The result of the decision tree classifier is a set of decision trees or decision rules for each category. This service method creates such a model from the document-term matrix created in the previous method. The sequential version builds the model for all categories and stores it in one file. The process of building the model iterates over a list of categories and for each of them creates a binary decision tree. The parallel version performs the same, but it distributes the work of building individual trees onto other services, so called workers, where partial models containing only trees of dedicated categories are created. These partial models are collected and merged into the final classification model.

In text-mining clustering processes we often have to deal with sequential tasks. Use of grid services can significantly speed-up process of text clustering. In practice choice of optional method or their parameters can be important in solution of current tasks. There is often need for running many experiments on same documents collection to achieve objective comparison of effectiveness and quality of created models. *Parallel Ensamble Learning* is optimization-like method based on the parallel running of several different clustering algorithms on same data collection (also with different settings – parameters of algorithms). The main goal is to find optimal model.

Particular sequential algorithms run parallel and independent on different nodes of grid architecture, so no one needs to be run separately. This simple process parallelization can be easily achieved by GUI in GRIDMINER and dynamic control of the grid tasks and jobs. On the level of Web layer user sets parameters

of clustering algorithms using GUI. These parameters are coded and together with other hidden parameters are stored in XML database. After calling of some clustering method particular instance can find appropriate parameters through specific key which is also used for distinction of parallel running services. In case of k-means two parameters were used – number of clusters and maximum number of iterations, for GHSOM implementation starting number of rows and columns of top layer SOM were used as well as minimal number of instances for creation of new hierarchy map.

Services are implemented on top of the Globus Toolkit 3.0, which is a standard for developing Grid applications. In order to implement the text mining support in the GridMiner's GUI, the client was designed as a web service. Simple JSP (JavaServer Pages) page was designed and it serves as a simple configurator of the input parameters for the service. Input parameters include the document collection location and set of parameters selecting the weighting scheme. This client can be simply enhanced with other service configuration settings including the selection of filters etc [4].

7 Experiments

7.1 Tested Datasets

In experimental part of this three datasets were used for testing. At first we used the collection *Reuters-21578*[6] database, which is a standard database of text documents for text classification. We used the *ApteMod* split version, which consists of 9603 training documents in 90 categories.

The second used database is collection of *MEDLINE*[7] documents. In the field of biomedical research, the MEDLINE database at the National Library of Medicine has become standard platform for scientific investigation. It contains citations and abstracts of medical literature from science and research projects to administrative papers. *MEDLINE* contains references from articles about 3 600 specialized magazines at 70 countries of the world.

The third was small *Times60* dataset, only used for clustering. It is a collection of newspaper articles from the 1960's. It consists of 420 documents with different war and political themes about different countries like Russia, Great Britain, Malyasia, Egypt, etc.

[6] http://www.daviddlewis.com/resources/testcollections/reuters21578/
[7] http://medline.cos.com/

7.2 Classification of Textual Documents

The main aim of this part is to introduce a classification (categorization) of textual documents with using ability of GridMiner, as an infrastructure for regulation and realization processes of knowledge discovery, and indexing services, as well as JBOWL library, as a basic service for text mining [4].

The main goal of the experiments was to prove, that the implementation of text mining processes can significantly reduce the time needed to construct the classification model. The results of the experiments on the Reuters dataset are shown in the table (Tab. 1):

Tree Depth h	Time (s)		Precision (%)	
	Jbowl	GridMiner	Micro-averaged	Macro-averaged
5	649,890	10,880	83,29	63,54
10	635,953	5,093	84,34	61,46
20	777,391	9,130	83,38	61,32
25	786,422	13,737	83,36	61,31
30	614,969	13,253	83,09	61,29
40	787,094	14,140	82,94	61,28
50	785,938	8,763	82,91	61,28

Table 1
Results of decision trees algorithm on Reuters dataset

We started the experiments with the sequential version of the service, in order to compare it with the Grid version implemented in the GridMiner system. We can see that difference of time needed to build the classification model using sequential JBOWL version and grid version of the same algorithm is significant. Moreover, the experiments in the JBOWL library were executed in sequential fashion, so time needed to find the optimal classification model was rapidly increased. One of the most important advantages of the GridMiner system is possibility of parallel execution of different tasks including the building model task. That, in fact, allows users to find the optimal classification model in very short time. Experiments with the same settings were performed on the *MEDLINE – Ohsumed* database. The results are shown in the table (Tab. 2):

Tree Depth	Time (s)		Precision(%)	
	Jbowl	GridMiner	Micro-averaged	Macro-averaged
5	3758,593	48,551	65,62	37,79
10	4368,781	310,468	55,27	35,39
20	4276,688	182,695	52,39	35,12
25	4411,547	219,725	52,19	35,11
30	4359,015	364,919	52,14	35,11
40	4699,485	358,699	52,14	35,11
50	4816,156	379,7	52,14	35,11

Table 2
Algorithm of decision trees on Medline dataset

As we can see from these results, in case of very large datasets, working with pure sequential text mining algorithms can take very long time to finish them. The same set of experiments was executed in the GridMiner system using parallelized versions of the algorithms. In general, we can see, that using the grid platform can bring significant benefits, when working with extremely large datasets, in this case, the experiments in Grid platform were executed approximately 16,46 times faster than using sequential JBOWL library.

7.3 Clustering of Textual Documents

The main goal of first experiment was to find optimal value of *k* parameter in k-means algorithm. First tested dataset was Reuters with local computer and sequential running of different parameter's values. Quality of models was evaluated using mean square error objective function based on the variance of training examples according to appropriate centroids of clusters. Different values for *k* were used from interval 10-150 (with 10 as a step), best objective function was achieved for *k=120*.

Then same task was solved using GridMiner process workflow (prepared simply using GUI) to achieve significant benefits from Parallel Ensamble Learning (PEL) method. Of course, from the qualitative point of view, result is same optimal parameter *k=120*. Important fact is that local computer sequential running is in case of GridMiner usage replaced by the PEL-based running and whole process ends after finishing of longest runned instance of algorithm, e.g. in our experiment 0.89 seconds (sequential run of all experiments on local computer – 30 minutes). Same experiments were also provided with larger collections – Medline (but with different *k* parameter values – from 20 to 100, with step 20) and PEL method achieved results after 2 seconds (sequential run on local computer – almost 4 hours).

Similar experiments were realized also with GHSOM algorithm. Because local computer tests are very time consuming, we have used smaller dataset Times. Starting number of rows and columns of top layer map was 2, changing parameter was minimal number of instances for creation of new hierarchy level by expanding of neurons on actual layer, values were from 20 to 60 (with step 10). PEL method provided all models after 12 seconds (sequential run on local computer took approximately 3 minutes).

Our experiments proved that GridMiner can significantly (using PEL method and GUI-based possibility to predefine text clustering process workflow) improve process of clustering of textual documents in spite of using sequential implementation of clustering algorithms. Next step have to be implementation of parallel (distributed) version of algorithms itself. As it was proved in case of classification, this can then gain benefits from both possible effective grid-based

improvements – PEL for parallel running of differently parametrized algorithms and parallel (distributed) run of algorithm instances by oneself.

Conclusions

Integration of the text mining services into the GridMiner system opens ways to a plenty of various possibilities for building the distributed text mining scenarios. Using the Grid as a platform it is possible to access different distributed document collections and perform various text mining tasks. Service oriented architecture allows the extensions of the system simply by adding different services for various algorithms. These algorithms can run in parallel fashion with different settings. Another way how to use the Grid infrastructure is parallelization of the algorithms. Some methods such as induction of the decision trees can be simply modified.

Main subject of this article was to present the idea of modification and implementation text mining services into the distributed Grid environment. JBOWL is a library, which contains methods for preprocessing, classification, clustering and evaluation techniques. On the other hand, GridMiner is architecture for parallel and distributed data mining in the Grid environment. It is natural, that using the JBOWL library and its implementation in the GridMiner system can significantly reduce time needed to perform the experiments on the large textual data collections. Moreover, the Grid environment opens ways to plenty of various possibilities to modify various text mining algorithms into the parallel or distributed versions. The most significant benefit, that using the Grid platform offers, is maximum effectiveness with minimal time costs.

The main advantage of the GridMiner system is Graphical User Interface. Users can create various workflows according to their needs. Different algorithms can run in parallel fashion, and these algorithms can be parallelized or distributed.

Acknowledgement

The work presented in this paper was supported by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic within the project No. 1/4074/07 ‘Methods for annotation, search, creation, and accessing knowledge employing metadata for semantic description of knowledge’ and project No. 1/3135/06 ‘Methods and tools for design of the integrated distributed applications based on ambients – higher-level agents’, and by the Slovak Cultural and Education Grant Agency within the project No. 3/3124/05 ‘Virtual laboratory for management of supply-purchase strings’.

References

[1] Foster, I., Kesselman, C.: Computational Grids, The Grid – Blueprint for a new Computing Infrastructure, Morgan Kaufmann (1999)

[2] Bednar, P., Butka, P., Paralic, J.: Java Library for Support of Text Mining and Retrieval. In Proceedings of Znalosti 2005, pp. 162-169, Stará Lesná (2005)

[3] Bednar, P., Butka, P.: JBOWL – Java Bag-of-Words Library. In: Proc. of the 5th PhD Student Conference, Technical University, Košice (2005)

[4] Sarnovský, M.: Textmining in the Grid Environment. In: Znalosti 2006: 5. ročník konference, VŠB-TU Ostrava (2006)

[5] Sarnovský, M.: Textmining Services in the GRIDMINER System. In: WDA 2005: Workshop on data analysis, Elfa, Košice (2005)

[6] Brezany, P., Janciak, I., Woehrer, A., Tjoa, A.M.: GridMiner: A Framework for Knowledge Discovery on the Grid - from a Vision to Design and Implementation. Cracow Grid Workshop, Cracow, December 12-15 (2004)

[7] Berkhin, P.: Survey of Clustering Data Mining Techniques. Accrue Software, Inc. (2002)

[8] Dittenbach, M., Merkl, D., Rauber, A.: Using Growing Hierarchical Self-Organizing Maps for Document Classification. In Proc. of European Symposium on Artificial Neural Networks – ESANN 2000, Bruges, April 2000, Belgium, ISBN 2-930307-00-5, pp. 7-12