Analytical Platform Based on Jbowl Library Providing Text-Mining Services in Distributed Environment

Martin Sarnovský(✉), Peter Butka, Peter Bednár, František Babič, and Ján Paralič

Faculty of Electrical Engineering and Informatics,
Department of Cybernetics and Artificial Intelligence,
Technical University of Košice, Letná 9/B, 042 00 Košice, Slovakia
{martin.sarnovsky, peter.butka, peter.bednar,
frantisek.babic, jan.paralic}@tuke.sk

Abstract. The paper presents the Jbowl, Java software library for data and text analysis, and various research activities performed and implemented on top of the library. The paper describes the various analytical services for text and data mining implemented in Jbowl as well as numerous extensions aimed to address the evolving trends in data and text analysis and its usage in various tasks reflecting the areas such as big data analysis, distributed computing and parallelization. We also present the complex analytical platform built on top of the library, integrating the distributed computing analytical methods with the graphical user interface, visualization methods and resource management capabilities.

Keywords: Text and data mining · Software library in java · Data preprocessing · Web portal

1 Introduction

Question of integrated analytical solutions has become interesting in recent years to improve the end-users orientation in wide range of available services, methods, algorithms or tools. The aim was to bring these services closer to the non-expert users and provide the possibilities to use them without deep knowledge about their implementation details or internal modes of operation.

The work presented in this paper represents our activities in building of the coherent and complex system for text mining experimental purposes built upon the distributed computing infrastructure. Such infrastructure can offer computational effectiveness and data storage facilities for proposed on-line analytical tool that comprises of various services for knowledge discovery in texts and provides specific data and computing capacity. Our main motivation is to provide coherent system leveraging of distributed computing concepts and providing simple user interface for users as well as administration and monitoring interface.

Text mining [1] aims at discovery of hidden patterns in textual data. For this topic, there is available a textbook [2], which we wrote in Slovak for our students. It describes

I. Khalil et al. (Eds.): ICT-EurAsia 2015 and CONFENIS 2015, LNCS 9357, pp. 310–319, 2015.
DOI: 10.1007/978-3-319-24315-3_32

the whole process of knowledge discovery from textual collections. We describe in details all preprocessing steps (such as tokenization, segmentation, lemmatization, morphologic analysis, stop-words elimination), we discuss various models for representation of text documents and focus on three main text mining tasks: (1) text categorization [3, 4]; (2) clustering of textual documents [5, 6]; (3) information extraction from texts [7, 8].

Finally, we describe service-oriented view on text mining and present also selected distributed algorithms for text mining. Second part of the textbook [2] is devoted to description of our Jbowl (Java bag of words library) presenting its architecture, selected applications and a couple of practical examples, which help our students easier start for practical work with Jbowl on their own text mining problems. In this paper we want to present the latest advancements in Jbowl library, which makes it usable also for big data text mining applications and invite broader audience of the World Computer Congress to use this library in various text mining applications.

2 Concept of Analytical Library

2.1 Jbowl

Jbowl[1] is a Java library that was designed to support different phases of the whole text mining process and offers a wide range of relevant classification and clustering algorithms. Its architecture integrates several external components, such as JSR 173 – API for XML parsing or Apache Lucene[2] for indexing and searching.

This library was proposed as an outcome of the detailed analysis of existing free software tools in the relevant domain [9]. The motivation behind the design of this library was existence of many fragmented implementations of different algorithms for processing, analyses and mining in text documents within our research team on one hand side and lack of equivalent integrated open source tool on the other hand side. The main aim at that time was not to provide simple graphical user interface with possibility to launch selected procedures but to offer set of services necessary to create the own text mining stream customized to concrete conditions and specified objectives. The initial Jbowl version included:

- Services for management and manipulation with large sets of text documents.
- Services for indexing, complex statistical text analyses and preprocessing tasks.
- Interface for knowledge structures as ontologies, controlled vocabularies or lexical WordNet database.
- Support for different formats as plain text, HTML or XML and various languages.

These core functionalities have been continuously extended and improved based on new requirements or expectations expressed by researchers and students of our department. Detailed information can be found in [10] or [11].

[1] Basic Jbowl package - http://sourceforge.net/projects/jbowl/.

[2] https://lucene.apache.org/.

The second main update of the library offers possibility to run the text mining tasks in a distributed environment within task-based execution engine. This engine provides middleware-like transparent layer (mostly for programmers wishing to re-use functionality of the Jbowl package) for running of different tasks in a distributed environment [12]. In the next step, new services for aspect-based sentiment analysis or Formal Concept Analysis – FCA (cf. [13]) were added [14] to extend application potential of the library in line with current trends. In case of FCA subpart related to processing of matrices from Jbowl BLAS (Basic Linear Algebra Subprograms) implementation was used and extended in order to work with FCA models known as generalized one-sided concept lattices [15] and use it for other purposes like design and implementation of FCA-based conceptual information retrieval system [16]. There is also extension of services related to processing of sequences within the data sets and processing of graph-based data, which is partially based on Jbowl API and its models.

2.2 Services for Distributed Data Analysis

Our main motivation was to use the problem decomposition method and apply it in data-intensive analytical tasks. Distributed computing infrastructure such as grid or cloud enables to utilize the computational resources for such kind of tasks by leveraging the parallel and distributed computing concepts. There are also several existing frameworks available offering different methods of parallel/distributed processing using the principle such as mapreduce, in-memory, etc. In order to support computation-intensive tasks and improve scalability of Jbowl library, we have decided to use GridGain[3] platform for distributed computing.

Jbowl API was used as a basis for particular data processing and analytical tasks. We decided to design and implement distributed versions of classification and clustering algorithms implemented in Jbowl. Currently implemented algorithms are summarized in Table 1.

Table 1. Overview of currently implemented supervised and unsupervised models in Jbowl.

	Sequential	Distributed
Decision tree classifier	✓	✓
K-nearest neighbor classifier	✓	✓
Rule-based classifier	✓	
Support Vector Machine classifier	✓	
Boosting compound classifier	✓	✓
K-means clustering	✓	✓
GHSOM clustering	✓	✓

In general, the process of the text mining model (classification or clustering) creation is split into the sub-processes. As depicted in Fig. 1, one of the nodes in

[3] http://www.gridgain.com.

distributed computing infrastructure (master node) performs the decomposition of particular task (data or model-driven) and then assigns particular sub-tasks onto available resources in the infrastructure (worker nodes). Worker nodes produce partial outputs which correspond to partial models (on partial datasets). Those partial models are collected and merged into the final model on the master at the end. The concrete implementation of sub-tasks distribution is different in particular model types, we will further introduce the most important ones.

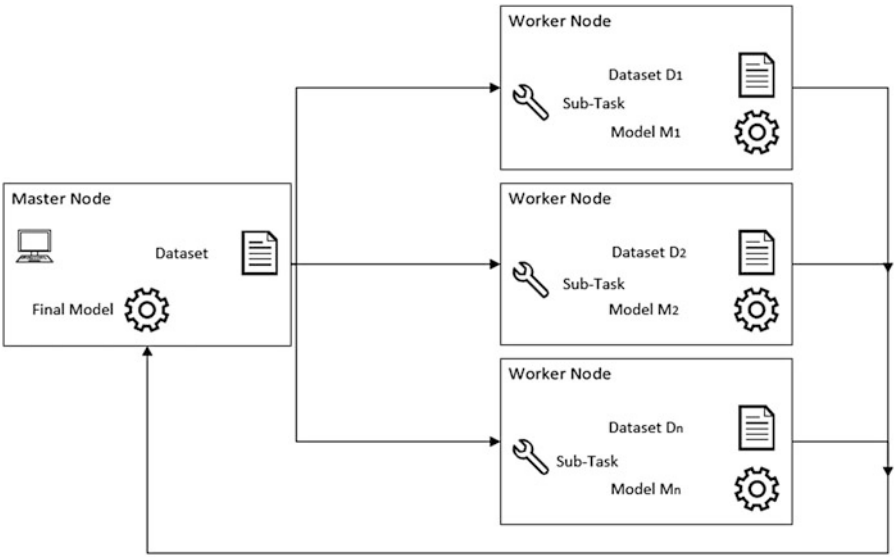

Fig. 1. General schema of the sub-task distribution across the platform

For induction of decision trees, Jbowl library implements generic algorithm where it is possible to configure various criteria for splitting data on decision nodes and various pruning methods for post-processing of the induced tree. Distributed version of the classifier [17] considers the multi-label classification problem, when the document may be classified into one or more predefined categories. In that case, each class is considered as a separate binary classification problem and resulting model consists of a set of binary classifiers. In this case, particular binary classifiers represent the sub-tasks computed in distributed fashion.

Our distributed k-nearest neighbor (k-NN) classification algorithm was inspired by [18]. In this solution we used the Jbowl k-NN implementation as a basis and modified it into the distributed version that split the input data into the chunks and calculates the local k-NN models on the partitions.

Another set of Jbowl algorithms modified into the distributed versions were clustering ones. Distributed implementation of GHSOM (Growing Hierarchical Self-Organizing Maps) [19] implementation uses MapReduce (GridGain implementation) paradigm and is based on parallel calculation of subtasks, which in this case represents

the creation of hierarchically ordered maps of Growing SOM models [20]. Main idea is parallel execution of these clustering processes on worker nodes. Distributed version of K-Means clustering algorithm is based on methods presented in [21, 22]. Our approach separates the process of creation of k clusters among the available computing resources so the particular clusters are being built locally on the assigned data.

The FCA algorithms are in general computationally very expensive when used on large datasets. This issue was solved by decomposition of the problem. Starting set of documents were decomposed to smaller sets of similar documents with the use of clustering algorithm. Then particular concept lattices were built upon every cluster using FCA method and these FCA-based models were combined to simple hierarchy of concept lattices using agglomerative clustering algorithm. This approach was implemented in distributed manner using the GridGain, where computing of local models was distributed between worker nodes and then combined together on master node.

Further, we have implemented specialized FCA-based algorithms of generalized one-sided concept lattices using the Jbowl API for sparse matrices and operations with them, which are able to work more efficiently with sparse input data usually available in text-mining and information retrieval tasks. Here we have provided experiments in order to test ratios for computation time reduction of sparse-based implementations in comparison to the standard algorithms [23]. Then, distributed version of algorithm for creation of generalized one-sided concept lattices was designed, implemented and tested in order to show additional reduction of computation times for FCA-based models [24]. The extended version of experiments was realized with real textual datasets [25], which proved behavior of previous experimental results on reduction of computation time using mentioned distributed algorithm for generalized one-sided concept lattices.

Also, we have currently finished implementation of selected methods (classification and clustering) provided in portal-based way which are able to run tasks for experiments defined by user in BOINC-based infrastructure. BOINC[4] is well-known open-source platform for volunteer distributed scientific computing. In this case Jbowl package is in the core of the system and is used for running of text mining experiments defined by user setup. These experiments are decomposed to BOINC jobs, pushed to BOINC clients and result of their distributed computations is returned back to server and provided to user [26].

The vision of the whole system is to re-use computational capacities of computers within university laboratories for volunteer-based computation. Our system has potential to support researchers to start their experiments and use additional cloud-like features of distributed computing using BOINC-based infrastructure. Currently, we have implemented also a graphical user interface which hides complexity behind creation of BOINC jobs for clients using dynamic forms and automation scripts for creation of jobs and analysis and presentation of the results provided to the user.

[4] BOINC - https://boinc.berkeley.edu/.

2.3 Services for Optimization

In some cases, the analytical processes can be complex, so our plan for the future development is to extend the system with the recommendations for less experienced users to improve their orientation in computing environment. These recommendations will be generated based on the observed patterns how other users are using the system and generating the results. For this purpose we will use our analytical framework designed and developed within KP-Lab project[5]. The core of this framework includes services for event logging, logs storage, manipulation with logs, extraction of common patterns and visualization of event/pattern sequences [27].

Patterns can be understood as a collection (usually a sequence) of fragments, each describing a generalization of some activity performed by users within virtual environment, e.g. sequence of concrete operations leading to the successful realization of clustering analysis. The success of this method for generating recommendations based on actual user behavior in virtual computing environment strongly depends on the quality of collected logs. Extracted and visualized information and patterns can be used not only for recommendations generation, but also for evaluation of user behavior during the solving of the data analytical tasks.

Another kind of optimization methods were implemented on the resource usage level. As mentioned in previous sections, several models are deployed on and use the distributed computing infrastructure. Main objective of these optimization methods is to improve the resource utilization within the platform based on type of performed analytical task as well as on the dataset processed. Several methods were designed for that purpose.

In general, system collects the dataset characteristics, including the information about its size and structure [28]. Another kind of data is collected from the infrastructure itself. This information describes the actual state of the distributed environment, actual state of the particular nodes as well as their performance and capacity. Depending on the type of analytical task, such information can be used to guide the sub-task creation and distribution. Sub-tasks are then created in order to maintain particular sub-task complexity on the same level and distributed across the platform according to the actual node usage and available performance and capacity.

2.4 Infrastructure Services

An important condition for the proper functioning and efficient of the presented services is a technical infrastructure providing necessary computing power and data capacity. We continuously build our own computing environment in which we can not only deploy and test our services, but we're able to offer them as a SaaS (Software as a Service). Simplified illustration of the used infrastructure is shown in Fig. 2.

Basic level contains several Synology network attached storages (NAS) with WD hard drives providing customized data storage capacity for various purposes; i.e. it is possible to use SQL or NoSQL databases or some types of semantic repositories. The

[5] http://web.tuke.sk/fei-cit/kplab.html.

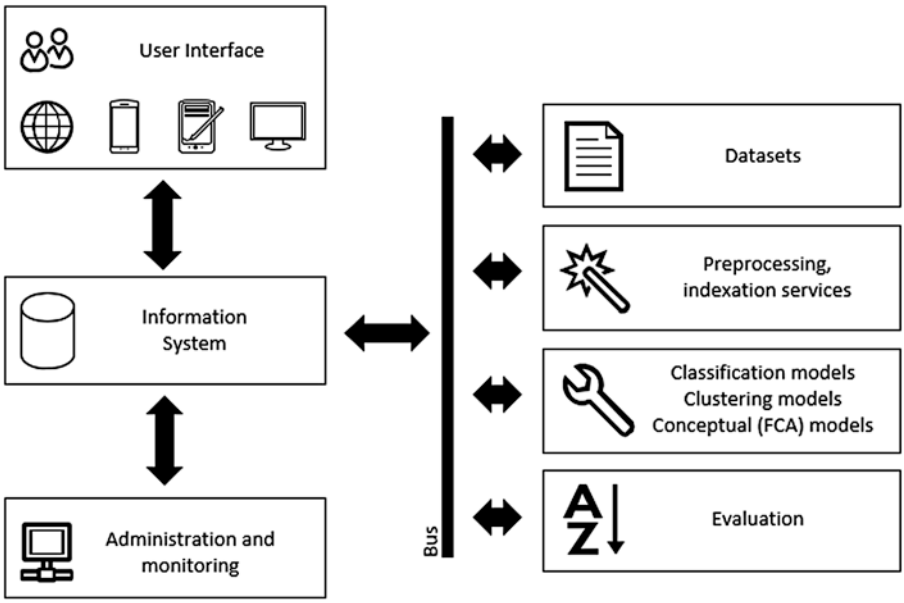

Fig. 2. Architecture of text mining analytical system

second level is represented by high-performance IBM application servers that are used for execution and data manipulation. This part of the infrastructure is separated and inaccessible to external users.

Graphical user interface with offered services and user functionalities is provided by web server and available for different end-user devices as traditional PC, laptops or tablets. Specific part of the deployed analytical system is the administration and monitoring module. Several modules are deployed on distributed computing infrastructure and interface to manage the platform itself is necessary. Such administration interface is implemented as a web application and enables to monitor the current state of the environment, including the operational nodes, their actual state, load, capacity as well as running tasks. If necessary, it is possible to disconnect the node from the platform, or add a new node, check their state and perform several actions to interact with them (stop halted tasks, free memory/space, check the network latency and restore the default configuration). Data collected using this module is also utilized in task distribution, as briefly described in Sect. 2.3.

On the other hand, people interested in our analytical and data processing services can download them, customize services based on their own preferences or requirements and finally deploy the customized platform on their own infrastructure.

Also, as it was written in Sect. 2.2, we have created some BOINC infrastructure from computers in our laboratories for students, which is able to provide additional computing capacity for BOINC-based applications. This paradigm is known as virtual campus supercomputing center and BOINC is widely used by several universities in the world in order to get some more computational capacities from their computers within campuses. After completion of testing phase we would like to provide graphical

user interface for more researchers to run Jbowl experimental tasks, for which particular models are computed on BOINC clients and returned to user. In the future it can be interesting to find interoperable connection in usage of cloud-based infrastructure defined above and volunteer-based BOINC infrastructure under one common platform, e.g., where capacities of both parts are managed together in order to achieve more efficient analytical services.

3 Conclusion

The need for software libraries for support of text mining purposes is not new, but their importance is increasing because of new requirements arising in the era of big data. An important factor is the ability of the existing products to respond to the changes in the areas as IT infrastructure, new capacity opportunities for parallel computing, running the traditional text and data mining algorithms on the new infrastructures, development of the new algorithms for data processing and analysis using new computational possibilities and finally the design and implementation of simple understandable and easy to use user environment.

Basically, presented work seeks to respond to all of these challenges and address them in the practical output of the relevant research and implementation activities. The presented library does not try to compete with the other available text mining platforms, but rather represents the output of our continuous work. The developed system presents an ideal platform for our ongoing research as well as for the education. Presented tools are used by the students and teachers in teaching tasks; serve as the platform for numerous master theses and regularly being used in data analytical and research activities.

Acknowledgment. The work presented in this paper was partially supported by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic under grant No. 1/1147/12 (40 %); partially by the Slovak Cultural and Educational Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic under grant No. 025TUKE-4/2015 (20 %) and it is also the result of the Project implementation: University Science Park TECHNICOM for Innovation Applications Supported by Knowledge Technology, ITMS: 26220220182, supported by the Research & Development Operational Programme funded by the ERDF (40 %).

References

1. Feldman, R., Sanger, J.: The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press (2007)
2. Paralič, J., Furdík, K., Tutoky, G., Bednár, P., Sarnovský, M., Butka, P., Babič, F.: Text Mining (in Slovak: Dolovanie znalostí z textov). Equilibria, Košice, p. 184 (2010)
3. Sebastiani, F.: Machine learning in automated text categorization. J. ACM Comput. Surv. (CSUR) **34**(1), 1–47 (2002)
4. Machová, K., Bednár, P., Mach, M.: Various approaches to web information processing. Comput. Inf. **26**(3), 301–327 (2007)

5. Sarnovský, M., Butka, P., Paralič, J.: Grid-based support for different text mining tasks. Acta Polytechnica Hungarica **6**(4), 5–27 (2009)
6. Rauber, A., Pampalk, E., Paralič, J.: Empirical Evaluation of Clustering Algorithms. J. Inf. Organ. Sci. **24**(2), 195–209 (2000)
7. Sarawagi, S.: Information extraction. J. Found. Trends Databases **1**(3), 261–377 (2007)
8. Machová, K., Maták, V., Bednár, P.: Information extraction from the web pages using machine learning methods. In: Information and Intelligent Systems, pp. 407–414, Faculty of Organization and Informatics Varaždin (2005)
9. Bednár, P., Butka, P., Paralič, J.: Java Library for Support of Text Mining and Retrieval. In: Proceedings of Znalosti 2005, pp. 162–169, VŠB TU Ostrava (2005)
10. Butka, P., Bednár, P., Babič, F., Furdík, K., Paralič, J.: Distributed task-based execution engine for support of text-mining processes. In: 7th International Symposium on Applied Machine Intelligence and Informatics, SAMI 2009, 30–31 January 2009, Herľany, Slovakia, pp. 29-34. IEEE (2009)
11. Furdík, K., Paralič, J., Babič, F., Butka, P., Bednár, P.: Design and evaluation of a web system supporting various text mining tasks for the purposes of education and research. Acta Electrotechnica et Informatica **10**(1), 51–58 (2010)
12. Butka, P., Sarnovský, M., Bednár, P.: One Approach to Combination of FCA-based Local Conceptual Models for Text Analysis - Grid-based Approach. In: Proceedings of the 6th International Symposium on Applied Machine Intelligence, pp. 31–135. IEEE (2008)
13. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999)
14. Butka, P., Pócsová, J., Pócs, J.: Design and implementation of incremental algorithm for creation of generalized one-sided concept lattices. In: 12th IEEE International Symposium on Computational Intelligence and Informatics, CINTI 2011, pp. 373–378. IEEE, (2011)
15. Butka, P., Pócs, J.: Generalization of one-sided concept lattices. Comput. Inform. **32**(2), 355–370 (2013)
16. Butka, P., Pócsová, J., Pócs, J.: A proposal of the information retrieval system based on the generalized one-sided concept lattices. In: Precup, R.-E., Kovács, S., Preitl, S., Petriu, E.M. (eds.) Applied Computational Intelligence in Engineering. TIEI, vol. 1, pp. 59–70. Springer, Heidelberg (2012)
17. Sarnovsky, M., Kacur, T.: Cloud-based classification of text documents using the Gridgain platform. In: 7th IEEE International Symposium on Applied Computational Intelligence and Informatics, SACI 2012, pp. 241–245 (2012)
18. Zhang, C., Li, F., Jestes, J.: Efficient parallel kNN joins for large data in MapReduce. In: Proceedings of the 15th International Conference on Extending Database Technology (EDBT 2012), pp. 38–49. ACM, New York (2012)
19. Ditttenbach, M., Rauber, A., Merkl, D.: The Growing Hierarchical Self-Organizing Map, In: Proceedings of International Joint Conference on Neural Networks, Como (2000)
20. Sarnovsky, M., Ulbrik, Z.: Cloud-based clustering of text documents using the GHSOM algorithm on the GridGain platform. In: IEEE 8th International Symposium on Applied Computational Intelligence and Informatics, SACI 2013, pp. 309–313 (2013)
21. Joshi, M.N.: Parallel K-means Algorithm on Distributed Memory Multiprocessors, Project Report, Computer Science Department, University of Minnesota, Twin Cities (2003)
22. Srinath, N. K.: MapReduce Design of K-Means Clustering Algorithm. In: proceedings of International Conference on Information Science and Applications, (ICISA) 2013, pp. 1-5 (2013)
23. Butka, P., Pócsová, J., Pócs, J.: Comparison of Standard and Sparse-based Implementation of GOSCL Algorithm. In: 13th IEEE International Symposium on Computational Intelligence and Informatics, CINTI 2012, Budapest, pp. 67–71 (2012)

24. Butka, P., Pócs, J., Pócsová, J.: Distributed Version of Algorithm for Generalized One-Sided Concept Lattices. In: Zavoral, F., Jung, J.J., Badica, C. (eds.) IDC 2013. SCI, vol. 511, pp. 119–129. Springer, Heidelberg (2013)
25. Butka, P., Pócs, J., Pócsová, J.: Distributed Computation of Generalized One-Sided Concept Lattices on Sparse Data Tables. In: Computing and Informatics, vol. 34, no. 1 (2015) (to appear)
26. Butka, P., Náhori, P.: Using BOINC software to support implementation of research and project text mining tasks (in Slovak: Využitie softvéru BOINC pre podporu realizácie výskumných a projektových úloh dolovania v textoch). In: Proceeding of the 9th Workshop on Intelligent and Knowledge oriented Technologies, WIKT 2014, Smolenice, Slovakia, pp. 22–26 (2014)
27. Paralič, J., Richter, Ch., Babič, F., Wagner, J., Raček, M.: Mirroring of knowledge practices based on user-defined patterns. J. Univ. Comput. Sci. **17**(10), 1474–1491 (2011)
28. Sarnovský, M., Butka, P.: Cloud computing as a platform for distributed data analysis. In: 7th Workshop on Intelligent and Knowledge Oriented Technologies WIKT 2012, November 2012, Smolenice, pp. 177–180 (2012)