UDK 004.42:004.822
Preliminary communication

SOME APPROACHES TO TEXT MINING AND THEIR POTENTIAL FOR SEMANTIC WEB APPLICATIONS

Jan Paralič, Marek Paralič
Technical University of Košice, Košice, Slovakia
{Jan.Paralic, Marek.Paralic}@tuke.sk

Abstract: *In this paper we describe some approaches to text mining, which are supported by an original software system developed in Java for support of information retrieval and text mining (JBowl), as well as its possible use in a distributed environment. The system JBowl*[1] *is being developed as an open source software with the intention to provide an easily extensible, modular framework for pre-processing, indexing and further exploration of large text collections. The overall architecture of the system is described, followed by some typical use case scenarios, which have been used in some previous projects.*

Then, basic principles and technologies used for service-oriented computing, web services and semantic web services are presented. We further discuss how the JBowl system can be adopted into a distributed environment via technologies available already and what benefits can bring such an adaptation. This is in particular important in the context of a new integrated EU-funded project KP-Lab[2] *(Knowledge Practices Laboratory) that is briefly presented as well as the role of the proposed text mining services, which are currently being designed and developed there.*

Keywords: *Text mining, semantic web, service-oriented computing, web services, trialogical learning.*

1. INTRODUCTION

Our research and educational goals in the area of text mining and information retrieval with the emphasis on advanced knowledge technologies for the semantic web resulted in identification of the following requirements, which ideal software system for our purposes should possess.

1. Be able to efficiently pre-process potentially large collections of text documents with a flexible set of available pre-processing techniques.

[1] http://sourceforge.net/projects/jbowl
[2] http://www.kp-lab.org

2. Particular pre-processing techniques should be well adopted for various types and formats of text (e.g. plain text, HTML or XML).
3. Text collections in different languages were envisaged, e.g. English or Slovak, as very different sorts of languages require significantly different approaches in pre-processing phase.
4. Support for indexing of and retrieval in these text collections (and experiments with various extended retrieval techniques).
5. Well-designed interface to knowledge structures such as ontologies, controlled vocabularies or WordNet.
6. Easy composition of various text pre-processing and text mining methods, tasks or algorithms.

The decision to design and implement a new tool, Java library for support of text mining and retrieval, was based on the detailed analysis of existing free software tools that could be used to support the abovementioned functionality requirements.

We found four different groups of tools:

- Text indexing and retrieval tools (such as e.g. Lucene [1]),
- Tools for text processing (e.g. GATE [2], JavaNLP [3]),
- Tools and APIs for support of the process of knowledge discovery in databases (Weka [4], KDD Package [10], JDM API [5]),
- Frameworks for work with ontologies (e.g. KAON [7]).

Each of the group covers very well one or two from the requirements stated above, but none of the tools supports all the requirements and therefore they are not suitable for intelligent support of applications that need to use text mining and semantic retrieval tasks.

Proposed Java library for support of text mining and retrieval called *JBowl* provides an easy extensible and easy to learn modular framework for pre-processing and indexing of large text collections, as well as for creation and evaluation of supervised and unsupervised text-mining models.

The rest of the paper is organized as follows. Next, section 2 briefly describes main supported text-mining tasks. Section 3 provides some use case scenarios, how the system has been exploited in a real system. But what is needed in order to adopt *JBowl* in such a way that a distributed application could profit from available text mining techniques? Section 4 briefly summarizes current state of the art in available technologies for the service-oriented computing and web services.

Section 5 presents a new integrated project KP-Lab, where the requirements for text mining services have arisen. Section 6 provides a brief sketch of current design of the text mining services to be used within the distributed KP-Lab platform and finally, Section 7 concludes with summarising the main advantages of the proposed approach to intelligent text mining and semantic retrieval in distributed applications.

2. TEXT MINING USING *JBowl*

JBowl has the same architecture as the standard Java Data Mining API (JSR 73 specification [5]). This architecture has three base components (application programming interface - API, text mining engine - TME, and mining object repository - MOR) that may be implemented as one executable or in a distributed environment. Figure 1 depicts architecture of the *JBowl* (Java Bag Of Words Library) implementation. The implementation has three-tier architecture with the data stored in a separate repository (i.e. in relational/object database). The API is a set of user-visible classes and interfaces that allows access to services provided by the TME. TME provides the infrastructure that offers a set of text mining services to its API clients. TME can be implemented as a local library or as a server of a client-server architecture. The TME uses a MOR, which serves to persist text mining objects.

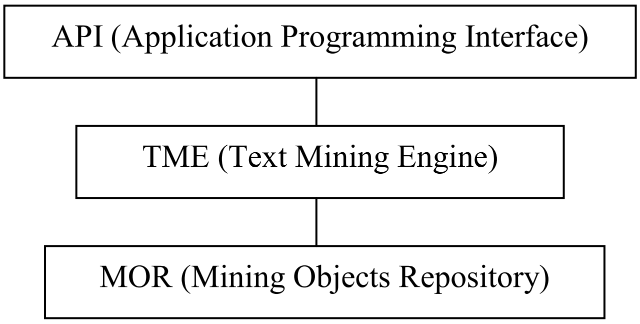
API (Application Programming Interface)
TME (Text Mining Engine)
MOR (Mining Objects Repository)

Figure 1. *JBowl* three-tier architecture

Reference *JBowl* implementation supports the following tasks.

2.1. DOCUMENT ANALYZING

Text analyzing on some language level is required by all following text-mining tasks. For analyzing task *JBowl* provides tokenizer classes, which divide text into tokens and various token filters, including filters for token normalization, stemming, collocations and stop words filtering. Some other token filters can provide more sophisticated processing like POS tagging or word sense disambiguation.

2.2. COMPUTING STATISTICS ON TEXT DATA

This task computes term and category occurrence and co-occurrence statistics required for building of various text-mining models.

2.3. BUILDING A MODEL

JBowl currently enables users to build classification models, clustering models and attribute selection models. To build models, users define settings that describe the type of the model, selected algorithm, and input data (i.e. training data set and validation data set used for evaluation of the model in learning phase). After a model is built by the TME (Text Mining Engine), it can be persisted in the MOR (Mining Object Repository).

Settings provided by the users can be divided into two levels: general settings related to model *function* (i.e. classification, clustering or attribute selection) and settings related to selected *algorithm* (i.e. Naive Bayes classifier, SOM neural networks (Self Organizing Maps), or information gain attribute selection).

JBowl provides a set of common Java classes and interfaces that enable integration of various classification methods. The design of *JBowl* distinguishes classification algorithms (i.e. SVM, linear perceptron, etc.) and classification models (i.e. linear classifier, rule based classifier). There are many possibilities how to implement selected algorithm. For example, the implementation of SVM algorithm is simply a wrapper around the SVMlib library. In this way, other packages like Weka can be integrated. Algorithms can even be implemented in different programming languages (i.e. C, C++) and integrated into *JBowl* with Java Native Interface [11].

2.4. TESTING A MODEL

Model testing provides estimation of accuracy for predictive (classification) models. Test task accepts a model and data for testing. As a result of model testing, *JBowl* can produce accuracy measures such as micro and macro averaged recall and precision.

2.5. APPLYING A MODEL

Applying a model to a new document produces one or more predictions or assignments. For supervised text mining, model applying produces predictions of category assignments possibly along with their corresponding probabilities. If unsupervised text mining (i.e. clustering) is applied, a document is assigned to a cluster. *JBowl* enables batch scoring for a collection of new documents as well as single document scoring, intended for real-time response.

3. USE CASE SCENARIO

The Webocracy[3] project addressed the problem of providing new types of communication flows and services from public institutions to citizens, and improving the access of citizens to public administration services and information. The new types of services increase the efficiency, transparency and accountability of public administration institutions and their policies toward citizens.

Within this project a *WEBOCRAT* system[4] has been designed and developed [9]. *WEBOCRAT* system is a Web-based system comprising Web publishing, computer-mediated discussion, virtual communities, discussion forums, organizational memories, text data mining, and knowledge modelling. The *WEBOCRAT* system supports communication and discussion, publication of documents on the Internet, browsing and navigation, opinion polling on questions of public interest, intelligent retrieval, analytical tool, alerting services, and convenient access to information based on individual needs.

WEBOCRAT intelligent retrieval mechanism is built on top of *JBowl* functionality. Document analysis, indexing, vector representation and API for text mining have been used as a basis. Three different text-mining tasks have been experimented for their possible exploitation within or for the *WEBOCRAT* system [8]. *Clustering* and *association rules* mining are used to support development and maintenance of the ontology. But the most important is use of *text categorisation* support for semi-automatic annotation of newly added text resources as well as for automatic routing of users' informal submissions to particular department.

[3] IST-1999-20364 Webocracy: "Web Technologies Supporting Direct Participation in Democratic Processes"

[4] http://www.webocrat.sk/

Moreover, *full-text retrieval* mechanism has also been added and tightly integrated with the concept-based retrieval. That means, if the user starts e.g. with full-text search, in addition to the results of the full-text query he/she will get also a list of relevant concepts and clicking on them invokes switch to concept-based retrieval.

3.1. DESIGN AND MAINTENANCE OF THE ONTOLOGY

Clustering does not fit the functionality of the *WEBOCRAT* system, because documents in *WEBOCRAT* system are primarily organized by their links to knowledge model so that primarily knowledge model is used for document retrieval and topic-oriented browsing. On the other hand, it is useful to use techniques like GHSOM [6], because of its hierarchical structure that is tailored to the actual text data collection, *as a supporting tool within the initial phase, when the knowledge model of a local authority is being constructed*. This is true in such a case when local authority has a representative set of text documents in electronic form available for this purpose. It is assumed that these documents will be later on published using the *WEBOCRAT* system and linked to the knowledge model for intelligent retrieval purposes.

But users must be aware of the fact, that GHSOM does not produce any ontology. It is just a hierarchical structure, where documents are organized in such a way that documents about similar topics are topologically close to each other, and documents with different topics should be topologically far away from each other. Particular node in this hierarchical structure is labelled by (stemmed) words – terms, which occur most often in cluster of documents presented by this node. This list of terms can provide some idea about concept(s), which can be (possibly) represented in the designed knowledge model.

Finally, particular documents represent leave nodes of this hierarchical structure. It is in our opinion necessary to look carefully through the whole structure, including particular documents in order to make reasonable conclusions about particular concepts proposed for the knowledge model and relations among them.

3.2. SEMI-AUTOMATIC ANNOTATION OF DOCUMENTS

In the *WEBOCRAT* system, ontology is used as a knowledge model of the domain, which is composed from concepts occurring in this domain and relationships between these concepts. Information is stored in the system in the form of mainly text documents, which are annotated by set of concepts relevant to the document content.

One strategy for document retrieval is based on concepts. User selects interesting concepts and asks for information related to them. The decision about document relevance to the user query is based on the similarity between the set of query concepts and the set of concepts that are annotated to the document. This task of document retrieval can be viewed as a classification task when the decision is made, whether the document is relevant for the user or not. With appropriate ontology which models the domain well, use of this knowledge model can yield better results than e.g. retrieval based on vector representation of documents.

Retrieval accuracy depends on the quality of document's annotation(s). *Data mining methods can be very useful to guide user at annotating new document*. Annotation of the new document is a classification task *(text categorization task)* when we need to make decision which concept (concept represents category) is relevant to the content of the document. Webocrat system based on JBowl functionality does it semi-automatically, i.e. based on the text analysis of a new document, the system proposes to the user a ranked list of concepts that might be appropriate for annotation. User can add or delete some associations between new document and concepts, and these changes can be immediately integrated into classifier. This requires ability of incremental learning. Relevance weighting of the concepts to the new document is better than simple binary decision. Concepts are ordered by weight of the relevance to the new document and user can search for additional relevant concept(s) according to this ordering.

4. SERVICE ORIENTED ARCHITECTURE AND WEB SERVICES

If we want to make text-mining services a part of more sophisticated systems or just available for utilization from other systems over a network, one of the challenges today is to build them according to the Service Oriented Architecture (SOA). Services are means for building distributed applications and are used to build flexible service-based applications. If we build a distributed system according to SOA, the cornerstone is the loose coupling of software components. Such a system than consist of a set of collaborating software components (or services) with well defined interfaces, which accomplish a set of tasks by exchanging of messages. Service-oriented computing is a general topic, more specifically in the context of WWW technologies we are speaking about Web Services.

As defined by the World Wide Web Consortium [15], a Web Service is a software system identified by a URI (RFC 2396), whose public interface and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols. Web Services rely on the functionalities of publish, find, bind [16] and the components of a Web service Model include Service Providers, Service Broker and Service Requester. Web services are defined by their interfaces in particular about how they describe their functionality, how they register their presence, and how they communicate with other Web services. People who want to use Web services could connect to the Universal Description Discovery and Integration (UDDI) centre to search for required services. The information about the Web services described by Web Service Description Language (WSDL) can be acquired. And the user could use the Simple Object Access Protocol (SOAP) to transfer the requirement specification and receive the real service.

4.1. SEMANTIC SERVICE-ORIENTED COMPUTING

As described above, core Web Service technologies (WSDL, UDDI) define formal interface contracts, describing the message syntax, but do not address the semantics of those interfaces. This means, that the meaning of the exchanged data is not formally described. Since the emerging Semantic Web and Web Services have a similar target audience, namely application clients, therefore they are intended for automated processing and share a common base technology (XML), it seems to be straightforward to apply semantic web techniques to web services. The Resource Description Framework (RDF) is particularly intended for representation of metadata about web resources in general and web services in particular and represents a notation to express structured metadata which has also a XML-based format representation (RDF/XML).

On higher level of abstraction there are several web ontology languages which are based on RDF notation and are used for knowledge modeling, i.e. define vocabularies and express classes of information by means of the vocabulary and their inherent relationships.

The most applied ontology language today is probably OWL[5]. For the specific application field of web services, languages like OWL-S are evolving. These languages are derived from OWL and enable to define richer semantic for service specifications. This can enable richer, more flexible automation of service provision and use, and support the construction of more powerful tools and methodologies [17].

Another example is The Web services Modelling Language (WSML) [18] - family of languages which formalizes WSMO. The Web Service Modelling Ontology (WSMO) is a conceptual model for describing various aspects of Semantic Web services. Reference implementation of the WSMO is the Web Service Execution Environment (WSMX) [19]. The main goal of the WSMO as a combination of Semantic Web technologies and Web services is to add to Internet a new dimension. It should be not only an information repository for human consumption, but also world-wide system for distributed Web computing. Therefore, the WSMO is based on the following principles: Web compliance, ontology-based, strict decoupling of resources, centrality of mediation as a mean for dealing with heterogeneities, ontological role separation (user requests vs. available Web services), differentiation between the descriptions of Semantic Web services and executable technologies and formal execution semantic.

4.2. SERVICE COMPOSITION IN OPEN ENVIRONMENT

If we have available a set of independent services, the service composition is of great importance. It should be supported by every aspect of services – they can be described, selected, engaged, collaborated with and evaluated. Desirable abstractions for service composition in open environments should exploit opportunities offered by those services:

- because services are autonomous, we should not require them to be subservient to other services,
- because services are heterogeneous, we should develop expressive, standardizable representations (presently this is done for data, but not for processes and policies),
- because services are long-lived, evolving and operate in environments that produce exceptions, representations should handle such situations,
- because services can be cooperative, abstractions would represent how they behave in the awareness of the behaviours of other services.

Current approaches are connected to low-level invocation of services – they are not specially geared for enabling composition. Services are integrated through method invocation without regard to any higher-level constraints. Semantic web technologies make use of prior work done in workflow management systems, artificial intelligence approaches for planning, formal process models, multi-agent planning and description logic. The objectives for the development of Semantic Web services are to enable reasoning about Web services, planning compositions of Web services and automating the use of services by software agents. The goal is to make Web services unambiguously interpretable by a computer. Examples of ontologies used for description of Web services are already mentioned OWL-S or WSML. Using these ontologies, a Web service can advertise its functionality to potential users. A request for a service would then be matched against the Web service's advertisement via a matchmaking process, because the objectives of such a

[5] http://www.w3.org/TR/owl-features/

request are expressed as goals, which are high level descriptions of concrete tasks. Every requester expresses its goal in terms of its own ontology, which provides the means not only for human to understand the goal, but also for a machine to interpret is as a part of requester's ontology. Similar to the goal description, all semantic web services have their descriptions in their own ontology. In case the ontologies differ, the mediation process should be applied. Mediation is also utilized in case of heterogeneous communication protocols of involved services.

5. PROJECT KP-LAB[6]

KP-Lab (Knowledge Practices Laboratory) is an EU-funded integrated project, which started in February 2006 and will take five years. 22 partners from 14 different countries participate on this project. KP-Lab will create new theories, technological platform as well as pedagogical and professional methodologies aimed at facilitating innovative practices of sharing, creating and working with knowledge in education and workplaces. This is a very ambitious project in the area of technology-enhanced learning.

The project is based on the idea of trialogical learning. Trialogical learning [13] refers to the process where learners are collaboratively developing shared objects of activity (such as conceptual artifacts, practices, products) in systematic fashion. It concentrates on the interaction through these common objects (or artifacts) of activity, not just among people or within one's mind (see Figure 2).

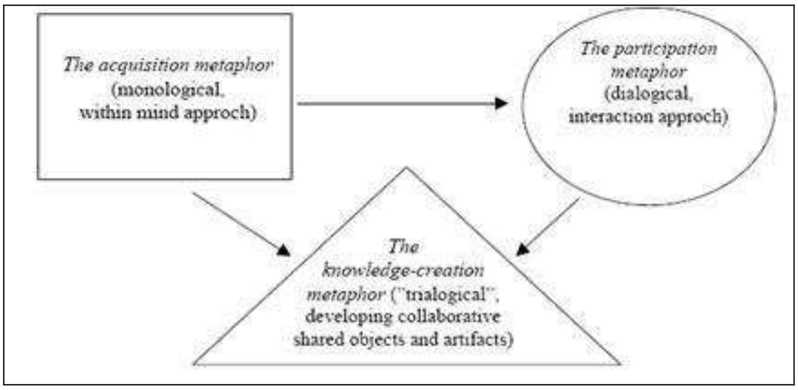
The acquisition metaphor
(monological,
within mind approch)
The participation metaphor
(dialogical,
interaction approch)
The knowledge-creation metaphor ("trialogical",
developing collaborative
shared objects and artifacts)

Figure 2. Three metaphors of learning **[13]**

The KP-Lab team from Technical university of Kosice (TUK) is involved in all technological workpackages, playing as such an important role in the technological aspect of the KP-Lab integrated research activities. Some of our tasks in this project are tightly related to text (or data) mining, e.g. semi-automatic annotation of various textual sources to concepts from various ontologies as well as for mining activity log files in order to support personalization, as well as discovery of new working patterns.

5.1. TOOLS TO BE DEVELOPED

Technological partners will design and develop the KP-Lab tools, which will be a cluster of inter-operable applications that include:

[6] http://www.kp-lab.org/

- a virtual collaboration space
- common tools for working with knowledge artefacts and for managing knowledge creation processes
- specific tools that will facilitate the discovery and exploitation of tacit and practice-related knowledge
- shared multimedia annotation tool
- ubiquitous cooperative conferencing and communication services
- a generic semantic web knowledge middleware for learning applications.

The KP-Lab technological framework will provide an operational technical architecture for KP-Lab tools and services, software modules allowing for interfacing KP-Lab tools with third-party software, as well as set of guidelines and reference documents to support the implementation.

5.2. KNOWLEDGE PRACTICES IN EDUCATION

Pedagogical partners will examine and model pedagogical practices related to collaborative teamwork where participants solve complex problems for real customers, e.g., enterprises, research communities, or public organizations. To support socialization into expert cultures already during studies, they design KP-Lab courses that involve crossing boundaries between educational and professional communities, in either actual or simulated contacts with professional knowledge practices.

The design experiments research will examine the role of KP-Lab tools in these processes, including knowledge creation, collaboration, argumentation, team training, and ubiquitous access to knowledge resources. Longitudinal research activities attempt to capture and trace individual and collective transformations that take place in higher education: engagement in sustained inquiry, development of agency, metaskills of collaboration, practices of knowledge work, and cultural learning.

6. TEXT MINING SERVICES IN DISTRIBUTED KP-LAB ENVIRONMENT

Text mining and extraction services are designed to assist users in creating or updating the semantic descriptions of KP-Lab knowledge artefacts. The semi-automatic generation of these descriptions or even of new KP-Lab ontologies relies on the textual information attached to particular artefacts.

Although they can be in various forms (e.g., textual documents, conceptual maps, video sequences, images, etc.), the knowledge artefacts often contain textual information directly in its content, or indirectly in textual annotations given by users. The textual description is analyzed using different text mining techniques. As a result of the text mining analysis, relevant concepts from the KP-Lab ontologies are suggested to the users during the formal description of knowledge artefacts. Moreover, unsupervised text mining techniques, such as clustering, can be used to find some unseen concepts (or clusters) in the set of analyzed textual resources. These may lead to, e.g., the suggestion to upgrade existing KP-Lab ontologies, as the knowledge of a user group evolves.

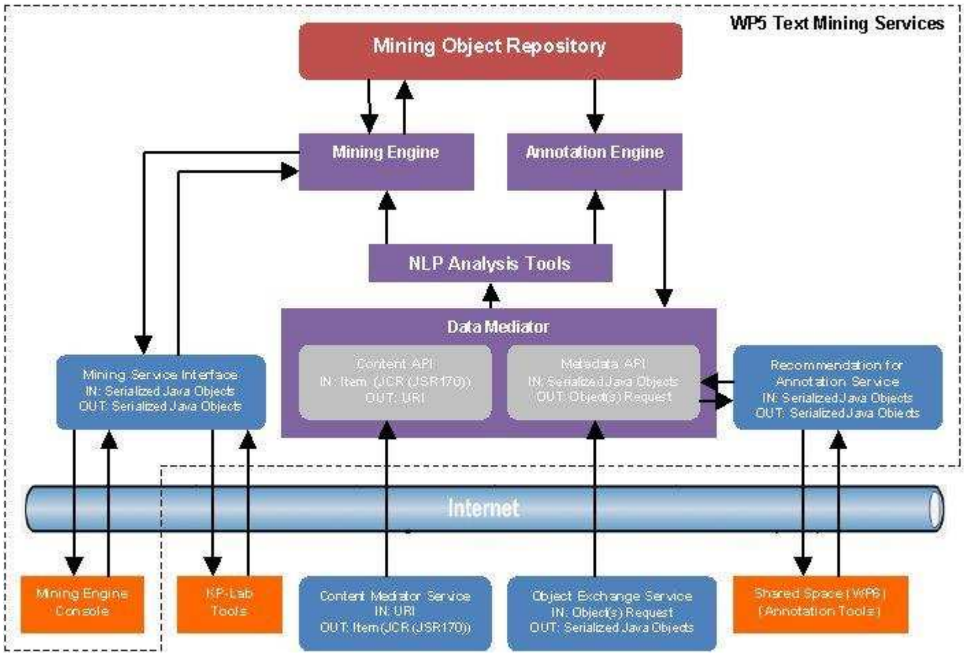

Figure 3. Text mining services in KP-Lab

The architectural design of the text mining services also follows the SOA principles. The structure of particular high-level services and its interconnections are presented on the Figure 3. The fundamental tasks for the envisioned services are *ontology learning and classification of knowledge artefacts.* Classification groups a given set of artefacts into predefined or ad hoc categories. Ontology learning automatically extracts significant terms from textual resources and converts them to a structure of concepts and their relationships. Functionality and algorithms used for ontology learning task is described in [20]. The classification task is presented in the following part.

6.1. CLASSIFICATION TASK

The classification task is proposed to automatically organize a set of knowledge artefacts into predefined or ad hoc categories. The predefined categories are the concepts of an existing ontology, which are chosen to semantically annotate the artifact. Ad-hoc categories are "new" concepts that can be identified in the textual descriptions and can potentially enrich the ontology structure. In this case, the task is close to the keyword extraction and summarization. This task can be used to create an initial dictionary for ontology from a set of "typical" artefacts.

The classification is supervised by a *model*, which is created from a training set of semantically already annotated artefacts. The model contains a set of parameters (weights, rules, etc. – based on the used algorithm) created in the process of training and used in the classification of unknown examples.

JBowl provides also a text categorisation method for active learning, which allows reducing the number of training examples required for efficient learning. Some implemented algorithms are based on simple heuristics that select examples according to the confidence of the classifier prediction for the given example. This heuristics do not require a validation set and can be used effectively to select a small set of labelled examples.

The following algorithms included in *JBowl* are considered to be used for classification: simple term matching, kNN, SVM, Winnow, Perceptron, Naive Bayes (multinomial and binomial), boosting, decision rules, and decision trees (various combinations of growing and pruning methods). The *JBowl* Java library [11] is built on the modular framework architecture (see section 2), which is highly extensible and supports SOA principles. Since it offers the required functionality for pre-processing, indexing and further exploration of text collections, it was chosen as a good candidate for implementing the classification tasks within the KP-Lab project.

For this purpose it will be necessary to provide all *JBowl* API classes and interfaces in form of semantic web services. In such a way particular tools will be able to search for a suitable text mining service in order to support various user tasks like annotation, creation of concept maps, analysis of textual contributions relevant to particular process, or knowledge artefact or to a user or a group. Moreover, providing *JBowl* functionality in form of semantic web services will open some new possibilities related to (semi-)automatic composition of suitable text pre-processing and text mining services.

Another prospective contribution that we pursue is in the field of flexible composition of available semantic web services that takes into the consideration semantic description of the service capability. The motivation is to achieve emergent functionality and more sophisticated functionality of a dynamically composed service without the phase of workflow design done by user or administrator statically. By restricting of the application area to the text-mining services, we can obtain valuable feedback after applying different approaches of the service compositions. Beside the process pursued in WSMX based on discovery, matchmaking, chaining and orchestration, for the dynamic service composition and execution we investigate also the possibility of using ambients - higher level agents.

7. CONCLUSIONS

In this paper we presented architecture and functionality of the *JBowl* library for support of text mining and information retrieval and its exploitation within a real client-server system called *Webocrat*. Furthermore, we summarised main ideas and technologies for service oriented computing and web services and sketched how *JBowl* can be extended in order to fit this kind of distributed architecture as well as advantages that such kind of implementation could bring. These ideas will be evaluated and further researched in the context of the EU funded integrated project KP-Lab, where the functionality offered by *JBowl* could be very useful and as we hope, could yield very interesting distributed experimental environment for composition of semantic web services (not only) in the context of text mining tasks. Extending of text-mining services with semantic descriptions and making them available as a set of semantic Web services will yield not only higher flexibility of text-mining services themselves. It enables also utilisation of these kinds of services by other distributed applications, which can express their needs as high level goals in appropriate ontology and contributes to the research in the area of dynamic service composition.

ACKNOWLEDGEMENTS

The work presented in this paper is supported by European Commission DG INFSO under the IST program, project KP-Lab, contract No. 27490 and by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic within the project No. 1/3135/06 “Methods and tools for design of the integrated distributed applications based on ambients – higher-level agents”.

The KP-Lab Integrated Project is sponsored under the 6th EU Framework Programme for Research and Development. The authors are solely responsible for the content of this article. It does not represent the opinion of the KP-Lab consortium or the European Community, and the European Community is not responsible for any use that might be made of data appearing therein.

REFERENCES

[1] Jakarta Lucene project, URL: http://lucene.apache.org/java/docs/features.html

[2] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, D.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, July 2002.

[3] Java NLP project: URL: http://www-nlp.stanford.edu/javanlp/, 2004

[4] Witten, I. H., Frank, E.: Data Mining: Practical machine learning tools with Java implementations, Morgan Kaufmann, San Francisco, 2000

[5] JSR 73: Data Mining API: URL: http://www.jcp.org/en/jsr/detail?id=73, 2004

[6] A. Rauber, D. Merkl, and M. Dittenbach: The Growing Hierarchical Self-Organizing Map: Exploratory Analysis of High-Dimensional Data. In: IEEE Transactions on Neural Networks, Vol. 13, No 6, pp. 1331-1341, November 2002.

[7] Staab, S. Studer, R.: An extensible ontology software environment, In *Handbook on Ontologies*, chapter III, pp. 311-333. Springer, 2004.

[8] Paralič, J. – Bednár, P.: Text Mining for Documents Annotation and Ontology Support. A book chapter in: “Intelligent Systems at the Service of Mankind” (W. Elmenreich, T. Machado, I.J. Rudas eds.), Ubooks, Germany, 2003, pp. 237-248

[9] Paralič, J. – Sabol, T. – Mach, M.: Knowledge Enhanced e-Government Portal. Proc. of the 4th IFIP International Working Conference on Knowledge Management in Electronic Government (KMGov 2003), Rhodes, Greece, May 2003, LNAI 2645, pp. 163 – 174

[10] Paralič, J. and Bednár, P.: *A Tool to Support of the KDD Process*. In Journal of Information and Organizational Sciences, Varaždin, Croatia, Vol. 27, 2003, pp. 15-27.

[11] Bednár, P., Butka, P., Paralič, J.: *Java Library for Suport of Text Mining and Retrieval*. In: Proc. of the 4th annual conference ZNALOSTI 2005, Stará Lesná, 2005, pp. 162-169

[12] Bednár P.: Active Learning for Text Categorization, In Proc. of the Znalosti 2006 Conference, Hradec Králové, Czech Republic, pp. 159-166

[13] Paavola, S., Hakkarainen, K.: “Trialogical” Processes of Mediation through Conceptual Artefacts, Technical Report for the KP-Lab consortium. University of Helsinki, Finland, (2006)

[14] Paralič M., Paralič J.: Data analysis for agent based mobile services, Journal of Information and Organizational Sciences, vol. 29, no. 1, 2005, ISSN 0351-1804, pp. 53-61

[15] Web Service Architecture Requirements, W3C Working Group, February 2004, http://www.w3.org/TR/2004/NOTE-wsa-reqs-20040211/

[16] Huhns, M.N.: Agents as Web Services. IEEE Internet Computing, 6(4), 2002, pp. 93-95

[17] Martin, D., et all: Bringing Semantics to Web Services: The OWL-S Approach, SWSWPC 2004, Lecture Notes in Computer Science 3387, 2005, pp. 26-42

[18] de Bruijn, J, Lausen, H., Krummenacher, R., Polleres, A., Predoiu, L., Kifer, M., Fensel, D.: The Web Service Modelling Language (WSML). Deliverable D16.1v02, WSML 2005, http://www.wsmo.org/TR/d16/d16.1/v0.2/

[19] Haller, A., Cimpian, E., Mocan, A., Oren, E., Bussler, Ch.: WSMX – A Semantic Service-Oriented Architecture, in Proceedings of the International Conference on Web Services (ICWS 2005), IEEE Computer Society Orlando, Florida, USA, 2005, pp. 321 – 328

[20] Smrž,P., Paralic, J., Smatana, P., Furdik, K.: Text Mining Services for Trialogical Learning. Paper accepted for the Czech-Slovak scientific conference Znalosti (Knowledge) 2007

Received: 18 March 2007
Accepted: 16 November 2007