

Classification and automatic concept map creation in eLearning environment

Karol Furdík¹, Ján Paralič¹, Pavel Smrž²

¹Technical University of Košice, Centre for Information Technologies,
Letná 9, 042 00 Košice, Slovakia
{Karol.Furdik, Jan.Paralic}@tuke.sk

²Brno University of Technology, FIT, Božetechova 2, 61266 Brno, Czech Republic,
University of Economics, FIS, W. Churchill Sq. 4, 13067 Praha, Czech Republic
smrz@fit.vutbr.cz

Abstract. The paper presents an approach to the classification of textual documents in a collaborative eLearning environment, as it is being applied in the context of the Knowledge Practices Laboratory (KP-Lab) project and further elaborated within the PoZnaT project. Classification and consequent semi-automatic annotation of learning materials as well as automatic concept map creation by means of text mining techniques and NLP methods enable users to organise and share the knowledge in a distributed learning system. Design principles and required functionality of the text mining services are discussed in the paper. The Mining Engine Console is presented as a tool for maintenance and administration of the classification and concept map creation processes. Finally, some initial examples of the classification on a real data and evaluation of the classification tool are described in this paper.

1 Introduction

Collaborative work and sharing of learning materials are important attributes of modern eLearning systems adopting knowledge acquisition as well as social participation metaphor [7]. Especially in the learning systems based on the emerging theory of Triological Learning, where the knowledge creation activities rely on usage, manipulation, and evolution of shared knowledge artefacts [8], the attributes of collaboration and knowledge sharing are even more emphasised. This is also the case of the Knowledge Practices Laboratory (KP-Lab) integrated FP6 EC project, focused on the development of a learning system aimed at “facilitating innovative practices of sharing, creating, and working with knowledge in education and workplaces” [2].

Various learning materials and resources, i.e. the knowledge artefacts, are semantically annotated by means of ontologies (taxonomies, concept maps, or domain ontologies) and are collaboratively investigated in the shared learning space of the KP-Lab eLearning system. The ontologies provide a conceptual framework for the descriptions of artefacts, as well as they define a structure on a domain of discourse. In the Triological Learning approach, the ontology itself is a subject of creation and evolution in the process of learning as a socially determined and interactive activity [8]. It implies

that the structure of knowledge artefacts, given by their semantic annotations, is constantly evolving to reflect the changes in the underlying ontology. To ease the process of ontology creation and semantic annotation for learners as well as to keep the consistency between ontologies and artefacts, the text mining and extraction services for clustering, ontology learning, and for classification were adopted within the KP-Lab project. Functional requirements for these services were specified according to the motivating scenarios defined by pedagogical partners of the project. Architecture and functionality of the text mining services was then proposed as a part of the Semantic Web Knowledge Middleware (SWKM) module of the KP-Lab system [9].

An adaptation of proposed KP-Lab eLearning system to the conditions of Slovak higher education is the main aim of the PoZnaT project [1]. As an enhancement of the text mining services, a suite of software NLP tools and a repository of language / linguistic resources for Slovak language, interconnected to ontology-based knowledge representations and semantic structures, will be developed within the project. The outcomes will be verified and evaluated on the pilot application within the course of Knowledge Management (this subject is taught within the study programs of Artificial Intelligence and Business Informatics at the Technical University of Košice).

This paper describes an adaptation of text mining services, namely classification and concept map creation techniques, for semantic annotation of knowledge artefacts in the context of the KP-Lab and PoZnaT projects. Section 2 presents basic features of the text mining, defined according to the scenarios and functional requirements formulated by users. Section 3 describes the architecture and functionality of the modules, including the signatures of provided API methods and the means of integration of this module to the whole eLearning system.

Section 4 gives an overview of design principles and features of the Mining Engine Console tool developed within the KP-Lab project. The Mining Engine Console is described as a web-based application for administration of the classification projects. The results of experiments carried out on real-world learning materials are presented and evaluated, together with suggestions for further improvements.

2 Text Mining on the Knowledge Artefacts

The classification of knowledge artefacts within the KP-Lab eLearning system is determined by the nature of these artefacts as learning objects that capture and preserve the knowledge shared within the community, as well as mediators of activities and tasks conducted by learners [8]. It is assumed that textual information is attached to the knowledge artefacts – directly in their contents, or indirectly in the textual descriptions given by users. Analysing these textual descriptions, the classification service supports a semi-automatic annotation of knowledge artefacts by organising them into pre-defined categories, i.e. into the concepts of an existing ontology.

The task of an automatic creation of concept maps identifies the most significant terms (representing concepts) and identifies relations among them. A set of artefacts provided by the user is processed first. The service can then identify concept candidates. The user can also specify a set of seed concepts and ask the service to find re-

lated concepts, as well as the type of the relation. The full concept map can be generated in the form of an RDF graph.

The knowledge artefacts are collaboratively created, managed, and investigated in the KP-Lab shared space. To define a practical usage of extended ontology manipulations in a real-world learning environment, and particularly to include the text mining services into the shared space properly, the motivating scenarios, use-cases, and high-level functional requirements were specified in co-operation of pedagogical and technical partners of the KP-Lab project.

2.1 Scenario for Semantic Annotation

The semantic annotation of knowledge artefacts according to a pre-defined classification ontology and consequent semantic search are required capabilities of the KP-Lab shared space, since they enable to share and exchange the information together with its semantic context (meaning) among learners. The following scenario was defined as main motivating scenario for the semantic annotation within the shared space by means of the classification services.

A student wants to create a new knowledge artefact for a final report document, using the “Create content item” form in the KP-Lab shared space (see Fig. 1). He or she uploads a document file into the shared space and specifies the metadata for the new artefact. The student can specify the metadata as the title, the description, as well as add one or more tags from a predefined vocabulary, to semantically annotate the new knowledge artefact (Fig. 1).

The screenshot shows a web form titled "Create a content item". It contains several sections:

- Title**: A text input field.
- Description**: A text input field.
- Choose content type**: A group of radio buttons for "Wikipage", "File", "Link", and "Discussion".
- Responsibility of**: A label followed by the text "Natalia Sobenina" and a checkbox.
- Add tags to content item**: A section containing:
 - A dropdown menu labeled "Select tag(s)" with the text "When typing the terms in list would highlight" and a plus button.
 - An input field labeled "Write an own tag" with minus and plus buttons.
 - "Send" and "Cancel" buttons at the bottom.

 Two callout boxes with arrows point to the "Add tags to content item" section. The first callout points to the dropdown menu and says: "This section allows to add semantic tags from the predefined controlled vocabulary." The second callout points to the plus button next to the dropdown menu and says: "This button opens drop-down menu with suggestions provided by the classification services."

Fig. 1. “Create new content” dialog with the semantic annotation

After the user has uploaded the document file, the shared space stores the file as a content item into the content repository and sends the content URL to the classification service. The classification service then requests the newly created content item from the repository, analyses its text content and/or structure, and applies a given

classification model. The result of the classification is a set of vocabulary terms, which will be provided to the user as a suggestion for further semantic annotation. Each term included in the result can contain additionally assigned real-valued score, denoting the confidence that the content should be annotated with the given term.

One possible way for representing the vocabulary terms within the shared space is to use a drop down menu (see Fig. 2). Suggestions for semantic annotations provided by the classification service are presented in a separate section of this menu. The user can browse the results ordered according to the confidence score, as well as to browse other terms of the vocabulary to supplement or correct the suggestions.

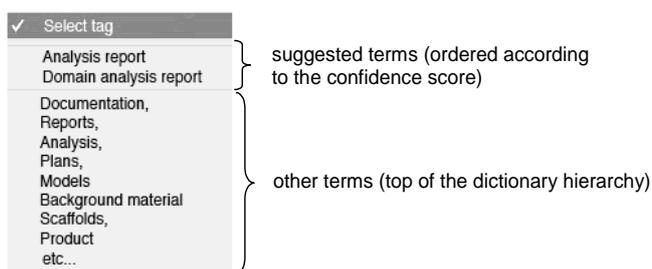


Fig. 2. Selection of terms taken from the PBL vocabulary

Various dictionaries can be specified as domain-specific conceptualisations for metadata tags, including dictionaries for document type as is specified in the Project-Based Learning (PBL) vocabulary. The PBL vocabulary was designed in [3] as a hierarchy of various types of items that are produced by learners during their collaborative work within the shared space, and describes activities related to the project management and production of end artefacts. The vocabulary contains a list of terms describing the annotation alternatives between the content items and defined tasks.

The above-presented scenario is an example of a single classification when only a single new artefact is classified to obtain the terms suggested for its semantic annotation. It is possible that a client of the classification service, i.e., the shared space application, sends a set of knowledge artefacts to be classified by the classification service. For example, the user can select a set of artefacts in his/her shared space and then use the classification service to additionally classify all these artefacts according to the selected controlled vocabulary. With this “batch” classification, the user can dynamically create different temporal views of his/her shared space. The result of the classification can then be permanently stored as a list of semantic annotations of the classified artefacts and can be used later to visualise the shared space.

2.2 Progressive Inquiry Collaborative Practice

To motivate the functionality of concept map creation services, let us present a pedagogical activity based on progressive inquiry type of collaborative practice where the starting points of the learning process are very open epistemological questions and written reflections [3]. These questions can be, for example:

- open-ended research questions about a certain topic or phenomenon;
- questions and reflections coming out of practical experiences, that are to be explored and explained by academic or theoretical knowledge integrated with practical knowledge.

The students upload all kinds of related materials to the Shared Space. This data is available for concept map creation services. The system analyses the provided resources and proposes a hierarchical set of concepts that can be extracted from the documents. After a while, the studied field begins to clear.

The instances of the previously structured set of concepts can be further modified. When the knowledge “database” grows, it might be useful to employ also clustering. The knowledge artefacts can be grouped according to a similarity measure. The resulting set of clusters can be explored manually, or become an input for the further processing by means of SWKM.

2.3 Functional Requirements for the Text Mining Services

Based on the scenarios for semantic annotation and progressive inquiry collaborative practice, the following functional requirements were specified for the clustering and concept map services:

- Pre-process documents (textual parts of the knowledge artefacts), produce an internal representation and store it into the Mining Object Repository. An URI is provided as an output, which can be subsequently used for accessing the data.
- Delete the pre-processed data from the Mining Object Repository, which will not be needed any more.
- Find clusters in the specified set of documents (URI to their representation in the Mining Object Repository)
- Identify concept candidates and rank them according to the estimated relevance, extract defining contexts for the terms
- Given a set of concepts, find related concepts from the documents provided by the user. Return a ranked list of candidate relations together with their types.
- Build the concept map and generate the RDF graph.

Following functionality was specified for the classification services:

- Creation of a training data set from already annotated knowledge artefacts to a pre-defined set of categories, i.e. concepts of existing domain ontology or terms of the PBL vocabulary. The textual descriptions of the artefacts are pre-processed and transformed into a term-document matrix. The classification service indexes the training data set and stores it into the Mining Object Repository.
- Creation of a classification model, based on the selected algorithm and on a given training data set.
- Modification (tuning) of the existing classification model, by changing the texts and/or categories in the training data set, as well as by editing the settings of the algorithm or switching to another algorithm.

- Provision of basic measures for existing classification model, e.g. by means of precision and recall. It should enable also to create, index, and store separate testing data set(s) (composed also from categorised artefacts) that can be used for more exact examination of the quality of the classification process.
- Verification and validation of the existing classification model. The model is no longer valid if a portion of training data set (e.g., the term-document matrix or the set of pre-defined categories) was modified. In this case, re-indexing of the model is needed to make it valid again.
- Classification of a set of unknown (not annotated) artefacts to the categories used for training. The output of this function is a set of weighted categories (concepts, terms) for each of the classified artefacts.

The proposed process of classification consists of the two phases: *creation and maintenance of a classification model*, including verification, evaluation, and modification of the model, and *usage of the model for classification* of unknown documents (artefacts). According to this division, the classification service is composed of two main sub-services: *LearningClassification service* and *Classify service*. Both sub-services use the *Mining Object Repository* [8] to save and retrieve classification models, text indexes, and other settings needed to perform the classification. Architectural design and functional specification of the classification service, as a component of the whole KP-Lab eLearning system, is described in the following section.

3 Architecture and Functionality of the Classification Module

The classification service of the KP-Lab system belongs to the Knowledge Match-Maker module, which provides services that support an advanced manipulation with the knowledge stored in the SWKM [9]. More specifically, the classification service together with pre-processing and NLP services, clustering and ontology learning services, belongs to the text mining and extraction services of the Knowledge Match-Maker module. The overall architecture of the SWKM services and the schema of text mining services were presented in [8].

To manipulate the textual descriptions of knowledge artefacts, both classification and ontology learning tasks use pre-processing and NLP analysis services. Implementation of the classification service is based on the JBowI Java library [4], providing a platform for several classification algorithms, tools for processing natural language texts, as well as some clustering techniques. For the purposes of clustering and ontology learning services, the GATE infrastructure [6] was adopted to provide means for tokenisation, stemming (or more sophisticated lemmatisation in morphologically rich languages), elimination of stop words, and POS (part-of-speech) tagging. Advanced NLP techniques such as chunking, WSD (word-sense disambiguation) or the full syntactic analysis will also be investigated for the text mining services.

The KP-Lab eLearning system is going to be adapted to the conditions of Slovak higher education within the PoZnaT project [1]. The main difference is in the processing of Slovak texts, where more advanced pre-processing and NLP analysis services are needed. Particularly, the modules for analysis on all the language levels and corre-

sponding data repositories (dictionaries, text corpora) are included as extensions of the existing KP-Lab system. The architecture of the text mining services enhanced by modules of analysis and processing of Slovak language is depicted on Figure 3.

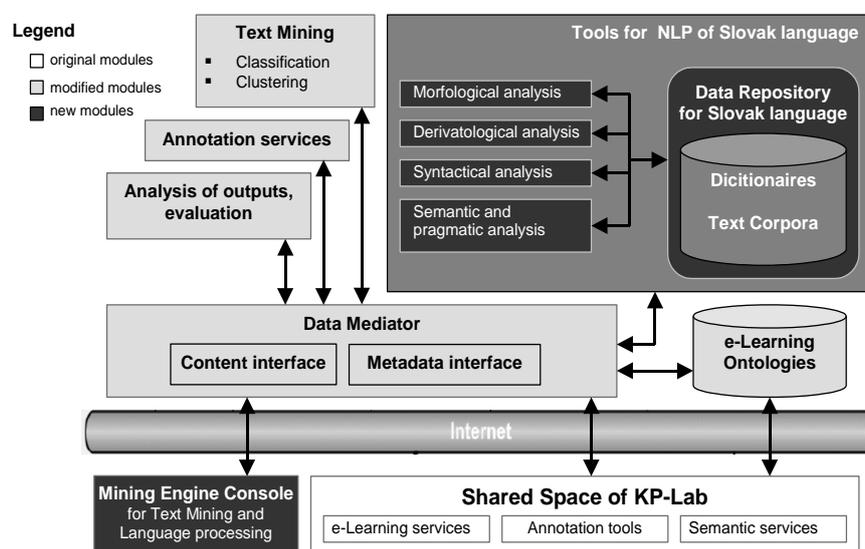


Fig. 3. Architecture of text mining services enhanced by Slovak NLP modules

The architecture of the SWKM and its components adopts the SOA (Service Oriented Architecture) principles. Following these principles, as well as the functional requirements outlined in Section 2.2 above, the signatures of API methods were designed for the sub-services of the classification service:

1. *LearningClassification* service provides the methods for creation, verification, modification, and removal of a classification model:
 - String **createModel**(String settings, String[] artefactURI)
 - String **verifyModel**(String modelURI, String[] artefactURI)
 - String **modifyModel**(String[] settings, String[] artefactURI)
 - String **deleteModel**(String modelURI)
2. *Classify* service provides the method for classification of artefacts:
 - String[] **classify**(String modelURI, String[] artefactURI)

The classification services itself have no user interface for these methods. It is assumed that the services are used in other KP-Lab tools, e.g. in the shared space (see scenario in Section 2.1), and the classification functionality will be used in the context of those tools. The only exception is the *Mining Engine Console*, envisioned as a web-based application that exposes classification (as well as some clustering) functionality for KP-Lab users. Design principles, provided functionality and implementation of the console application are described in the following section.

In parallel to the division of the classification service into two sub-services, the clustering and concept map services also distinguish two phases – pre-processing and

clustering/concept map creation proper. The signatures of API methods provided by the services are as follows:

1. *Pre-process* service is implemented as a pipeline of processing resources on top of the GATE engine. Additional language processing resources, that are necessary for the concept map service, integrate language-dependent tasks such as parsing, keyword extraction, co-occurrence statistics and semantic-distance computation:
 - URI **preprocess**(String[] artefacts, String[] seed)
 - void **delete**(URI preprocessedData)
2. *Clustering and concept map* services provide interfaces to unsupervised text mining methods:
 - String[] **findClusters**(String[] artefacts)
 - String[] **findConceptCandidates**(URI preprocessedData)
 - String[] **findRelatedConcepts**(URI preprocessedData, String[] concepts)
 - String[] **buildConceptMap**(URI preprocessedData)

4 Mining Engine Console

The Mining Engine Console is a web interface that provides an access to the text mining services for users. The console enables to manage the Mining Object Repository, to maintain classification models together with training and testing sets, view statistical reports for particular classification tasks, etc. First prototype of the Mining Engine Console (see Fig. 4) has been developed and tested on real learning materials.

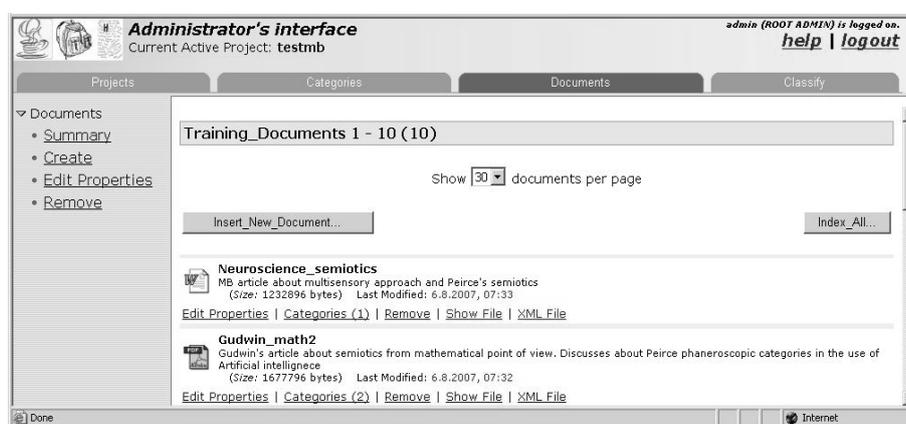


Fig. 4. Mining Engine Console. List of documents in the training set

The console is designed as a wrapper of the JBowl Java library [4], aiming to expose the JBowl text mining functionality through web interface. The prototype is implemented as a database client-server application, using the JSP technology and mySql database. However, next version of the console is envisioned to be able to access the core text mining services provided by JBowl via web services interface.

We plan to use the Mining Engine Console for several purposes. The main goal of this application is to maintain the classification models within the KP-Lab eLearning system [8], to support the semi-automatic annotation of knowledge artefacts in the collaborative learning environment. Second goal of the console is to visualise the services provided by the JBowl library, i.e. to serve as a reference application and also as a testing platform for JBowl. Third purpose is to provide means for pre-processing, NLP analysis, as well as classification and clustering services for Slovak texts within the PoZnaT project. In this case, as it is depicted in Figure 3, the console should be also capable to maintain the language/linguistic resources as text corpora, dictionaries and settings of NLP analysis modules. And finally, the Mining Engine Console should be used as a tool for students within the course of Knowledge Management at the Technical University of Košice.

4.1 Design Principles and Functionality of the Mining Engine Console

To fulfil all the envisioned usage purposes, the interface was designed in a general manner, providing all the basic functions to create and maintain classification processes. A project-based approach was applied in the console interface, where a single classification task is defined as a so-called “*classification project*”.

The classification project consists of:

- a selected classification algorithm and its settings;
- a set of pre-defined categories;
- a training set of documents, associated with the categories;
- a set of documents to be classified (a priori not associated with the categories, so called classification set);
- a testing set of documents, associated with the categories;

In the console prototype, the PDF, MS Word, HTML, and plain text formats of documents are supported and can be uploaded into the training, testing, and classification sets. The following classification algorithms are nowadays provided by JBowl library and 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). However, only the kNN and Perceptron algorithms are currently implemented in the prototype; we plan to extend the set of available classification algorithms, together with some clustering techniques (kMeans, SOM, GHSOM), in nearest future.

The console web application uses a role-based user access control. The roles as *administrator*, *trainer*, *tester*, and *guest user* were defined. The administrator defines and manages all the classification projects, the list of users and their roles, as well as the rest of system settings. The trainer is responsible for definition of categories and documents for a given classification project, for indexing the training set and creation of the classification model. The tester has the rights to use the model for classification of non-categorised documents and evaluates the classification by defining the testing set. Guest user has read-only access to the training and testing sets, he/she can use the existing model to classify non-categorised documents and view the output statistics.

The functionality provided by the Mining Engine Console web application can be described in the following steps:

1. *Definition of the classification project*, which includes selection of proper classification algorithm (currently, only the kNN and Perceptron algorithms are available in the prototype) and specification of its properties as TF-IDF scheme, thresholds, number of iterations, etc.
2. *Specification of categories* as a simple list of terms. The leaf nodes of the PBL vocabulary [3] were used as a basic category schema in the prototype.
3. *Creation of the training set* by uploading the documents in supported formats and by associating the documents with the categories. Plain texts are extracted from the documents and stored, together with category indexes, in the XML format required by the JBowl framework.
4. *Creation of the classification model*. This step includes indexing of the training set and creation of the term-document matrix. The algorithm with the properties specified in the classification project is then applied to build the classification model. The reports of term and category frequencies are automatically created and provided to users. The binary representation of the classification model is stored into the Mining Object Repository and the model is ready for its usage, i.e. for the classification of non-categorised documents.
5. *Evaluation of the model* by specifying the testing set of documents. The testing set is created similarly as the training set, by uploading the documents in supported formats and associating them with existing categories. The standard values of recall and precision are provided as a result of the evaluation.

In addition to these steps, the *validation* and consequent *re-indexing* of the classification model are important and required features, if the console is used in real learning environment. After the classification model is created, any changes in the training set or in the parameters of the classification algorithm will cause that the model will be no longer valid and needs to be re-built to be suitable for classification. The Mining Engine Console provides means to re-build the model semi-automatically; within the KP-Lab eLearning system, the Notification service [5] is designed to keep the consistency between classification models and artefacts annotated in the collaborative learning environment.

4.2 Testing and Evaluation

The Mining Engine Console prototype was tested on the three categorised sets of documents, obtained from pedagogical partners of the KP-Lab project:

- *EID UU*: University of Utrecht, The Netherlands. 138 PDF and MS Word documents classified into 5 categories. Language: Dutch.
- *UHE*: University of Helsinki, Finland. 56 MS Word and plain text documents classified into 5 categories. Language: English.
- *VSE*: University of Economics in Prague, Czech Republic. 97 PDF and MS Word documents classified into 7 categories. Language: English.

The documents were divided into training and testing data sets approximately in the 2:1 ratio. Training sets were indexed and classification models for kNN and Percep-

tron algorithms were created. Finally, the testing data sets were indexed and classified to evaluate the classification models. The recall and precision values obtained from this evaluation for the three test cases are presented in Table 1.

Table 1. Results of testing the Mining Engine Console on the three test cases. Second column contains the number of documents used in test case (all documents/training set/testing set), third column contains the number of classification categories.

| Test case | Documents All/Train/Test | Cat. | Algorithm | Recall [%] | Precision [%] |
|-----------|-----------------------------|------|------------|------------|---------------|
| EID UU | 138 / 90 / 48 | 5 | kNN | 70,4 | 27,8 |
| | | | Perceptron | 74,3 | 24,7 |
| UHE | 56 / 35 / 21 | 5 | kNN | 67,7 | 21,2 |
| | | | Perceptron | 65,4 | 18,6 |
| VSE | 97 / 65 / 32 | 7 | kNN | 84,0 | 42,3 |
| | | | Perceptron | 82,3 | 37,4 |

The best results were achieved in the VSE test case – it is most likely caused by higher number of categories, shorter extent (size) of the documents, as well as by proper distribution of documents into the categories. However, the results of all the test cases are influenced by the selection of classification categories as “Report”, “Essay”, “Email”, “End product”, etc. Such categories express the type of a document rather than its content. Assuming that the text mining methods are based on the analysis of the textual content, it is problematic to obtain better results when trying to classify into this kind of categories. However, significant improvement should be achieved by applying the NLP methods for supporting the creation of classification models, as it is envisioned in the near future within the PoZnaT project.

5 Conclusions and Future Work

The paper describes an adaptation of text mining services for semi-automatic support of semantic annotation of knowledge artefacts in the context of the KP-Lab project.

Currently (September 2007), the design of text mining services is mostly finalised. The prototype of the Mining Engine Console was developed and tested on real-world data. According to the work plan of the KP-Lab project, the implementation of first prototype should be ready by the end of 2007. The system will be tested in some of the KP-Lab pilot applications in spring 2008. In parallel, classification services together with the administration tool will be tested in summer semester 2008 on Technical University of Košice, within the lessons of Knowledge Management.

Use of text mining techniques in the context of a collaborative learning environment is a challenging task. To create the mining model properly, the global settings as a mining algorithm and its parameters need to be specified. Since the selection of proper algorithms is a specific feature of the text mining approach, it is required that this should be hidden from users. The provided solution should select the most adequate algorithms and its parameters automatically, according to the quantitative prop-

erties of the training set (as e.g., the number of artefacts, frequency and distribution of words in the textual content of artefacts, etc.). This is one of our future goals. Another one is the efficient adoption of the text mining services in (possibly frequently) changing space of knowledge artefacts in a collaborative learning environment.

Acknowledgements

The work presented in this paper was supported: by European Commission DG INFSO under the IST program, contract No. 27490; by the Slovak Research and Development Agency under the contract No. RPEU-0011-06; and by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic under the contract No. 1/4074/07.

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. Babič, F., Furdík, K., Paralič, J., Wagner, J.: Podpora procesov tvorby nových znalostí. DATAKON 2007, 20. - 23. October 2007, Brno, Czech Republic (2007). (in press)
2. Babič, F., Paralič, J.: Knowledge practices Laboratory (KP-Lab) Project. In: Proc. of the 6th annual conference Znalosti 2007, Ostrava, Czech Republic, 2007, pp. 364-367, ISBN 978-80-248-1279-3.
3. Bauters, M. et al: Semantic tagging according to PBL vocabulary requirements. Draft of requirements for semantic tagging in Shared Space M18 specifications. KP-Lab Internal Report of the KP-Lab project. (2007).
4. Bednár, P., Butka P., Paralič, J.: Java Library for Support of Text Mining and Retrieval. In Proc. from the Czech-Slovak scientific conference Znalosti (Knowledge) 2005, Stará Lesná, Slovakia (2005), pp. 162-169.
5. Belhaj Frej, H., Rigaux, Ph., Spyrtos, N.: User Notification in Taxonomy Based Digital Libraries (Invited Paper), ACM SIG-DOC Conference on the Design of Communication, Myrtle Beach SC, U.S.A. (2006), pp. 18-20.
6. GATE - General Architecture for Text Engineering - <http://www.gate.ac.uk>
7. Paavola S., Lipponen L., and Hakkarainen K.: Models of Innovative Knowledge Communities and Three Metaphors of Learning. In: Review of Educational Research, 74-4 (2004), pp. 557-576.
8. Smrž, P., Paralič, J., Smatana, P., Furdík, K.: Text Mining Services for Dialogical Learning. In: Proc. of the 6th annual conference Znalosti 2007, Ostrava, Czech Republic (2007), pp. 97-108, ISBN 978-80-248-1279-3.
9. Tzitzikas, Y. et al: Emergent Knowledge Artifacts for Supporting Dialogical E-Learning. Proc. of the TEL-CoPs'06: 1st International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice, Crete, Greece (2006), pp. 162-176.