Framework for Semi Automatically Generating Topic Maps

Lóránd Kásler[1] and **Zsolt Venczel**[1] and **László Zsolt Varga**[1]

Abstract. The amount of electronically stored textual information is continuously increasing both on the internet and in company assets, and there are no good solutions to easily locate the most needed information. Because search engines do not take into account the meaning of the word and its context, in the end the user has to select the right information from the unstructured result set. If the text is annotated and linked to the ontology of the annotation, then the user can directly navigate along the links of the semantic annotation to the desired information.

In this paper we present a software framework to semi automatically generate a semantic representation of the knowledge of the Networkshop conference series and display on a web portal the generated ontology together with the references to the occurrences of the instances in the source text. The framework presented in this paper makes advances in the following fields: we do not assume that the source text has uniform and formally defined structure, we address English and Hungarian text as well, we incorporate machine learning techniques in the process, and provide a flexible content management system for the presentation of the generated Topic Map on a web based portal.

1 INTRODUCTION

The amount of electronically stored textual information is continuously increasing both on the internet and in company assets, and there are no good solutions to easily locate the most relevant information. Although there are engines, like Google, for word based search, and the techniques are continuously improved, in the end the real search is completed by the user, because these search engines do not take into account the meaning of the word and its context. The semantic web tries to improve this by attaching semantic annotation to data and text. The semantic annotation is based on an ontology of the given domain. However the amount of information is huge and the semantic annotation cannot be done manually for large amount of text, therefore there is need for automated tools.

Once the information is semantically annotated, then the search can be improved in two ways. One way is that the search engine takes into account the semantic annotation of the text in order to further improve the result set of the search; the other is that there is no search engine and the user directly navigates along the links of the ontology of the semantic annotation to the desired information. The second approach has limitations towards large information sets like the whole internet, however in the scales of single portals or company information assets this can be a viable option. In addition the second approach has advantages as well. One advantage is that the ontology of the semantic annotation is closer to the thinking of the user and the user feels it much more comfortable to browse along the ontology than to select the right information from the unstructured result set of search engines. This holds in our case where we build a portal and a knowledge source specialised on a specific domain. Another advantage is that while navigating along the ontology of the annotation, the user may find other relevant and interesting information which he/she would not even think of.

Currently there are two main standards for representing the knowledge used for the annotation: the W3C standard [5] RDF/OWL and the ISO standard Topic Map [6]. We chose the Topic Map standard, because its concept is like and intelligent extension of the index of books with key features as topics, associations between topics, and occurrences of topics. We also chose the Topic Map standard, because it is very flexible in merging and extending different sets of Topic Maps.

In this paper we present a software framework to semi automatically generate a semantic representation of the knowledge and information present in a set of natural language text files, and display the generated ontology together with the reference to their occurrences in the source text on a web portal. The software framework is applied to semi automatically generate a Topic Map from ten years of the NetWorkshop conference proceedings [38]. The result is presented on a structured information portal and content management system.

The development of this software framework was motivated by the challenge of applying the Topic Map technology, the lack of such a framework, the lack of specialized and fast algorithm implementations with high precision on a medium data corpus. The implementation takes into account the specialities of the Hungarian language, mainly the problems of stemming.

The specific task, to semi-automate the construction of a Topic Map based on a conference was originally tackled by Steve Pepper and Lars Marius Garshol from Ontopia [1]. Their original intent was to describe a showcase on applying Topic Maps on real data rather than experimenting text mining algorithms and heuristics. The abstract concept of generating a Topic Map from any kind of semi-structured data is still an open field. Concrete techniques are mentioned in [2], or implemented in TMHarvest [3]. The Topic Map generating framework presented in this paper is an independent development from the above works and makes advancements in the following fields: we do not assume that the source text has uniform and formally defined structure, we address English and Hungarian text as well, we incorporate machine

[1] Computer and Automation Research Institute, Kende u. 13-17., Budapest, 1111 Hungary email:laszlo.varga@sztaki.hu

learning and information retrieval [18][19] techniques in the process, and provide a flexible content management system for the presentation of the generated Topic Map on a web based portal.

The structure of the paper is as follows: in Section 2 we summarize the technology we build on, in Section 3 we describe the framework that we developed for generating Topic Maps, in Section 4 we evaluate the framework and the generated Topic Map portal.

2 APPLIED TECHNOLOGIES

In this section we are going to summarize the technologies used for the development of the software framework. We used a broad spectrum of mature, open-source, Java technologies.

2.1 Topic Map

Topic Maps became an ISO standard in January 2000 [6], and the technology is in active development. Considered by some a rival, or a redundant specification for the WWC standard RDF [5], but the two specifications address different needs [9][10][17], and can coexist in several ways [4][8].

The key features of Topic Maps are: topics identified by their names; associations between topics; and occurrences of topics pointed to via locators. The key main advantages of this knowledge representation technology are data merging, Published Identities [14], rich set of metadata, and an element named "scope", which is mainly used for multilingual purposes [11][12][13].

There are many Open Source [33][34] and commercial implementations of Topic Map in Java, from Ontopia, Infoloom, Empolis and other vendors. There is even an effort to standardize the API used by vendors, called Topic Map API (TMAPI) [15].

2.2 Machine Learning in Java

For various analysis tasks the framework uses several machine learning and language processing techniques [20]. One of the most comprehensive architecture and collection of algorithms in this field is an open-source project of the University of Waikato, named Weka Machine Learning Project [21]. Besides broad variety of implemented classifiers, there are other, open-source extensions like jBNC [29].

Among several advanced algorithms, the framework contains a pluggable stemming package. We have successfully integrated a Hungarian language stemming software package, called Szószablya [27]. This way all other layers of the application dependent on stemming became language independent, because the abstract stemming package instantiates the needed sub package.

Although WEKA is one of the popular choices for machine learning and text mining tasks, we experimented with other frameworks such as YALE (Yet Another Learning Environment) [28] as well.

2.3 Ant Framework

The Ant Framework [16] is known as an open source build system, but besides being a modern replacement for make, its task oriented philosophy, easy configuration and integrated command line interface has a larger applicability. The main phases of the process implemented in our framework are modeled as Ant Tasks and can be controlled uniformly.

3 FRAMEWORK FOR GENERATING TOPIC MAPS

The framework for generating topic maps consists of a set of software tools and methods to support the execution of the process represented on Figure 1. The process has four phases: the data organisation, the analysis, the Topic Map population and the content management phase.

In the data organisation phase the raw source text available in various formats and structures is processed to have uniform structure. In this phase the metadata that can be extracted from the semi structure of the raw text is extracted and converted to a formal structure.

The goal of the analysis phase is to identify the main topics and their associations present in the source text. Two basic identification methods are applied. One is the identification of the topics and associations from the structure of the source text. For example topics like the paper title, the author, the affiliation of the author can be identified by pointing to the appropriate item in the structured metadata. We did not use named entity recognition, because we could not have defined associations between recognised entities easily. Associations like "a paper is authored by an author" can be identified by associating the items in the structured metadata. The other identification method is based on the analysis of the natural language text of the source text. Ideally this could be based on information retrieval methods to identify the topics and their associations mentioned in the papers. The implementation of the method on natural language understanding would have been too ambitious for our project, therefore we decided to use already existing external taxonomy or ontology to assign keywords to papers. The associations between keywords are defined by the external ontology. The result of the analysis phase is a Topic Map skeleton which is a combination of the external ontology and the ontology defined by the source text structure.

The Topic Map skeleton contains topic types which do not have occurrences. For example we know that there are authors and papers, the authors can write papers on different keywords, "Java Virtual Machine" and "operating system" are keywords, Windows and Linux are operating systems, and Java Virtual Machines can have implementations on different operating systems. However we do not know which authors wrote about which keyword and we do not know which papers contain which keyword. In the Topic Map population phase we identify the concrete instances of these topic types identified in the analysis phase. The result of the Topic Map population phase is a complete Topic Map of the source text.

The final phase of the framework is the content management phase. In this phase the completed Topic Map is loaded into an informational portal where the Topic Map can be presented to the user in a user friendly way using a content management system. With the help of the content management system the screen of the portal can be formatted and transitive associations can be added. For example if we know that authors write papers and papers are about keywords, then we can add the transitive association that authors write about keywords.

In the following we are going to detail the phases of the process of generating Topic Maps.

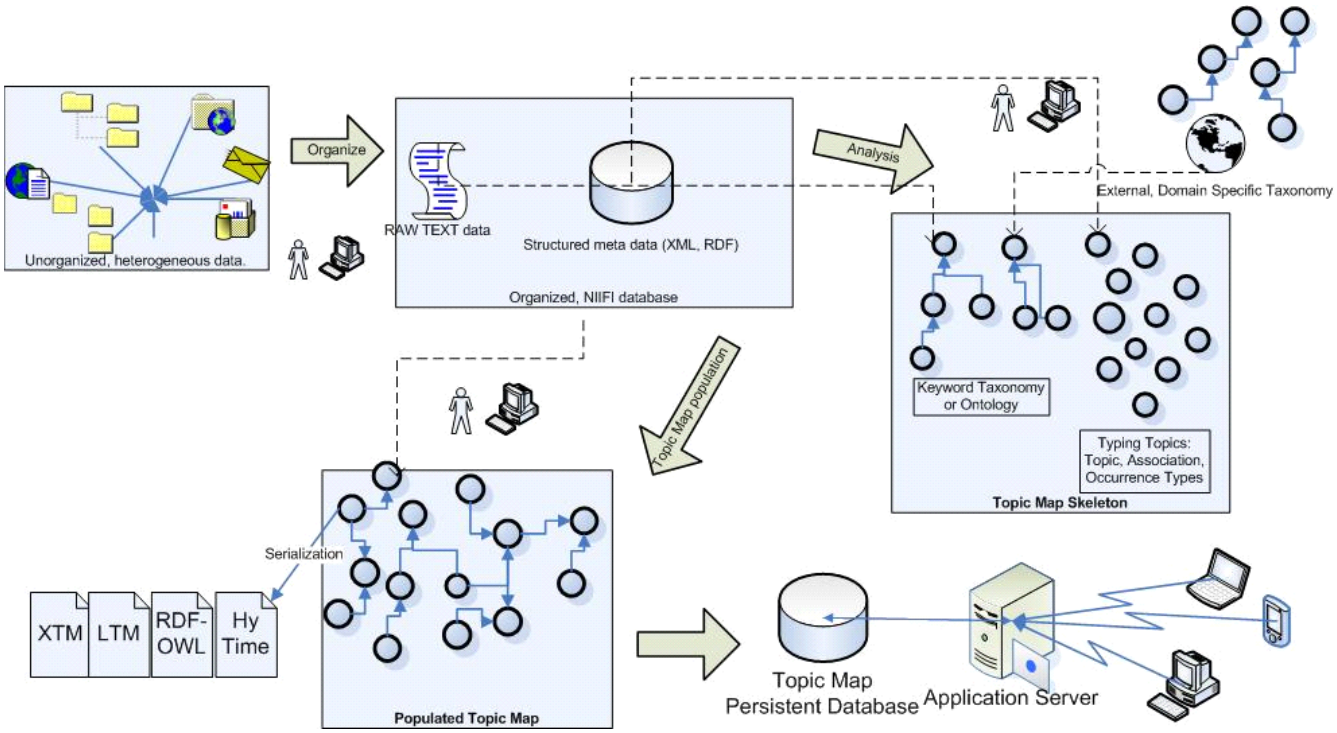

Figure 1. Framework for Generating Topic Maps

3.1 Phases of the Framework

Our solution consists of a semi-automatic system, capable of generating Topic Maps, from arbitrary complex data. It is a collection of tools, implemented in Java, forged together by a command line interface. The main design considerations were versatility, pluggability, runtime efficiency and incremental build. The project secondary objective, besides running a public portal, based on Topic Map technology, was to implement a Content Management backend for it. The backend leverages the used knowledge representation, thus it also has a Graphical Web interface for modifying the Topic Map. In order to achieve this, we had to maintain the incremental aspect of generating Topic Maps along all the tools.

The main task of generating the Topic Map is distributed over several runtime phases: Data Organization, Analysis, Topic Map Population. The Content Management Part of the Informational Portal based on the constructed Topic Map model, is in fact using the same architecture, to incrementally change underlying data. The generated Topic Map is persisted in a relational database, or in XTM [7], the XML interchange format for Topic Maps. The framework itself is agnostic of the chosen persistence alternatives, or the Topic Map engine, because it uses an abstract and standard API, called TMAPI.

As mentioned above these phases are incremental, which means, that they can augment any Topic Map model, indifferent of the used Topic Map engine and persistence technique. Most of the phases are semi-automatic, which means that user interaction is required to configure parameters or confirm certain assumptions made by several heuristic algorithms.

3.1.1 Data Organization Phase

Originally the data from the ten Networkshop conferences were in various formats and scattered in different places. Almost every conference had a different structure, or even worse: similar, but randomly discrepant directory trees. In this phase, we collected all the metadata from the data corpus and stored it in a structured way, using XML. The metadata to be extracted is identified by looking at the format of the papers and identifying for example that the first line is the title, the second is the authors list, etc. Several pattern matching techniques are used, such as regular expressions, to construct this preliminary database. The tools are manually configured to gather as much useful information as possible.

Another important part is textual data extractions from the different file formats used, because the conference paper formats changed from year to year. The parsers used are also Open Source implementations of parsers for the popular formats, such as Microsoft Word doc, PowerPoint, pdf and others.

3.1.2 Analysis Phase

As we said previously, the process of analysis leads to a Topic Map skeleton, containing the Typing Topics and Keyword topic instances.

The collection of typing topics in a Topic Map represents the ontology used by that model. Every topic is an instance of one of these typing topics. This ontology is the core, on which other tasks depend and it constitutes a solid base, on which layers of concrete data can be built. The process of discovering Typing Topics is also manually configured. Basically for every structured metadata

format the user has to create an XML configuration file containing the mappings. We used and enhanced the TMHarvest framework for this task. The mapping file contains several patterns, like XPath expressions, or Regular expressions to encapsulate the source of a typing topic. For the current data corpus we identified eleven typing topics, like Paper, Author, Conference and also other Association and Occurrence types.

The keyword topics are taken from an external ontology which may come from several sources. The actual implementation uses an external source, FOLDOC [24] to obtain a rich, domain specific ontology. Other taxonomies, web directories, or dictionaries could be easily used, such as ODP [25] and Babel [26]. The external source adapter system is customizable to create from virtually any format the desired keywords. The FOLDOC source is in a textual representation, which holds formatting metadata. The implemented parser for the FOLDOC text is based on several observations, which became rules. For example one of these rules is described as: a line starting with no trailing white spaces, and containing a few words represents a starting of a new keyword in FOLDOC. The associations between the other keywords are represented with special delimiters, for example a keyword is enclosed in parenthesis. These and other rules help the FOLDOC parser construct a true ontology represented in the Topic Map model.

Another approach to create the FOLDOC Topic Map representation would be to discover automatically the important keywords, phrases and associations between them, as in the case of the conference meta and textual data. Implementing this alternative is far beyond our project, but the current framework could stand as a basis for such an extension.

3.1.3 Topic Map Population Phase

The process of Topic Map Population is by far the most challenging and interesting task. It is configured the same way as the typing topics generator, but the used patterns are based on actual topic instances, like the instances of a Paper topic, or Author topic. The generating templates describe a mapping from every structured metadata record to the specified topic instance.

Even techniques based on a semi-structured or structured data face several morphological and semantic problems. The main problem is identifying the entities across several records. For instance the name of a person could be misspelled in a number of ways, or the order of the first and family name is not universal in many languages. Also the use of addressing like Phd., Dr., Msc. can be an obstacle for successful identification. We implemented several language dependent heuristics for tackling misspelling and other problems, but besides this there is also a special pattern file which encapsulates domain and data specific errors. This task uses a multi-phase approach and the heuristics are fired against the data model iteratively. Thus the underlying knowledge representation becomes more and more coherent after every pass.

Besides this first technical part of the populating process, which is based on metadata, there is another task which is based on the raw texts of the papers. These texts contain inherent associations not published explicitly through metadata. For instance the paper is associated to the categories described by keywords. Another example is that one author references another author in the text.

The automated document classification is implemented in a pluggable way. It can use several techniques from the field of unsupervised document classification and statistical information retrieval. To leverage the current implementations of such techniques, we integrated our tools with the WEKA framework.

The simplest approach to assign classifying keywords to papers would be a full text search based classification, for example by searching the abstract of each paper for the keywords of FOLDOC. This approach gives a heavily expanded classification, because every word occurrence is weighted equally. Although this gives rough estimation of used keywords, the final classification results are not acceptable. Using a pipeline architecture we managed to create chains of processing as shown on Figure 2. The original classification created by the simple search based approach is used, and refined in the second part. Using a Vector Space Model (VSM) of the papers, we managed to create a more accurate classification. The vectors in this model were the keywords found by the full text search and every paper could be represented as an element in this space, based on the relative relevancy of every occurring keyword in that particular paper. After conducting several classification experiments, we decided to create an association between the paper and a keyword if and only if the relative relevancy is in the first 66% among the other keywords present in the text of the paper. The magic number of 66% was decided intuitively: full inclusion was too much, half inclusion produced bad results, and the magic number seemed to be acceptable.

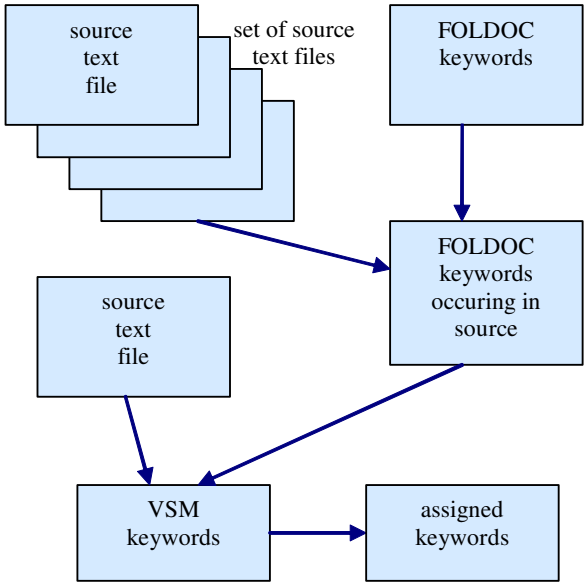

Figure 2. Chains of processing to find the most relevant topics in a paper

This combined technique is language independent, as far as the keywords and the words in the texts are correctly stemmed. Because FOLDOC was currently available in English, we used the English abstract of the papers. By constraining the FOLDOC keyword database to the subset found as a true occurrence in the conference papers, the translation is much easier.

3.1.4 Informational Portal / Content Management Backend

The tangible part, as far as the end user is concerned is the informational web-based portal. The main reason why we have chosen the Topic Map model to provide an abstraction for data representation is that the Topic Map representation of metadata and

the typing topics can easily be published on a portal. The core concepts of this portal are the topics and their associations. The user can navigate from one topic to another as in a Wiki from one page to another. From every type of topic all possible associations are visible and hyperlinkable. From a given author one can access all the authors, the conferences on which the author published and the keywords the author wrote about. By navigating to a selected keyword, all the associated papers are shown.

There are numerous generic frameworks to visualize and to publish topic maps. The Ontopia Omnigator [32] or the open-source TM4Web [35], like any other Topic Map application, are based on a Topic Map engine. The View part consists of easily modifiable Html templates, using a template engine like Jakarta Velocity [31] or standard JSP technology.

While the Topic Map engines fit well as a generic visualization, for the individual portals one must implement the whole navigation and view according to its design and concept. This is why we have chosen another approach to configure the whole portal through a content management backend by leveraging the representational metadata. This metadata is encapsulated in every typing topic and their templates. Content navigation and rendering is also based on typing topics.

In conclusion, the framework implemented on a Topic Map model is content management that leverages knowledge representation. It integrates several general editors that discover the actual type of modifiable data, and chose the appropriate template for the topic being edited or viewed. It uses the built in multi language support of the Topic Map paradigm, which was one of the priorities of this project.

As technology basis, we use the Tapestry web framework [22][23][30], which is a component oriented framework. We have implemented several generic components for every element of the Topic Map paradigm: the Topics, Associations, Occurrences and also lists powered by tolog queries [37] which are topic map queries similar to SQL, but having a Prolog like syntax.

We have extended this generic topic editing and managing framework to refine the specific tasks for specific topic types. Thus we implemented an interface which allows that a topic in administration mode is not only editable, but the user can perform specific tasks. The user is able to rerun the occurrence finder, the classifier or any other implemented action for the current type of topic.

4 CONCLUSION

This project tackled several technological and algorithmic challenges. It investigated the applicability of the Topic Map model on real data. We have experimented with different document classification algorithms and implemented a content management system based on this knowledge representation.

The framework for generating Topic Maps presented in this paper does not assume that the source text has uniform and formally defined structure, it handles English and Hungarian text as well, incorporates machine learning and information retrieval techniques in the process, and provides a flexible content management system for the presentation of the generated Topic Map on a web based portal.

At the time of the writing of this paper an initial Topic Map is generated and tested. The generated Topic Map contains 3537 topics, 723 papers, 973 keywords from ten years of Networkshop conferences. Manual annotation at this scale is not feasible, because the annotation is sometimes regenerated or incrementally extended at each year's conference. There are about eleven thousand keywords in FOLDOC, and in general we do not expect that the number of keywords would dramatically increase. The tools of the framework produce results in seconds when applied to the conference papers of the Networkshop series.

The project has proven the applicability of the Vector Space Model in categorization by reducing with an order of magnitude the irrelevant classifications and keywords. The deployed Topic Map portal is under test. Compared to the original conference web site, the Topic Map portal is user friendly and helps finding the relevant information in ten year's volumes of the Networkshop conference proceedings. The Topic Map generating process is semi automatic which allows the easy incorporation of coming volumes of the conference proceedings.

4.1 Future Work

A future improvement of the classification phase would be the usage of true learning based classifiers, such as Bayes classifiers or others alike. The occurrence or keyword discovery could be made directly from the textual data using advanced keyword and context extraction techniques. A viable solution would be integrating KEA [36], an open-source keyword extraction package, with the current framework.

Another, more visually appealing feature would be an interactive web-based or desktop GUI that guides the end user through the phases. The current Content Management system is generic enough, but it doesn't have yet the necessary abilities to create a full topic map from scratch. At least an ontology must be present in the model. To fully use the potential of Topic Maps, the internal portal metadata could be expressed in terms of Topic Map elements. Thus a generic editor could edit the system itself, if it is carefully configured.

ACKNOWLEDGEMENTS

We are thankful to the Hungarian National Information Infrastructure Development Program for participating in the project, specifying the requirements, giving advices and providing the source of the Networkshop conference series organised by them.

The framework presented in this paper was developed in the Topicportal project supported by the Hungarian Economic Competitiveness Operative Programme (GVOP AKF) under the GVOP-3.1.1-2004-05-0404/3.0 contract.

The project was initiated during the discussions we had with Steve Pepper and finally supported by Ontopia with a special license of the Ontopia Knowledge Suite for this project.

REFERENCES

[1] StevePepper, Lars Marius Garshol - The XML Papers: Lessons on Applying Topic Maps. http://www.ontopia.net/topicmaps/materials/xmlconf.html

[2] Geir Ove Gronmo - Automagic Topic Maps http://www.ontopia.net/topicmaps/materials/automagic.html

[3] TMHarvest http://www.folge2.de/topicmaps/tmharvest/userdoc01/en/index.html#features
[4] Lars Marius Garshol - Living with topic maps and RDF http://www.ontopia.net/topicmaps/materials/tmrdf.html
[5] Ora Lassila and Ralph Swick - Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation, 22 February 1999. Available from http://www.w3.org/TR/REC-rdf-syntax/
[6] ISO/IEC 13250:2000 Topic Maps, International Organization for Standardization, Geneva. Available from http://www.y12.doe.gov/sgml/sc34/document/0129.pdf
[7] Steve Pepper and Graham Moore (editors) - XML Topic Maps (XTM) 1.0, TopicMaps.Org. Available from http://www.topicmaps.org/xtm/1.0/
[8] Graham Moore - RDF and TopicMaps: An Exercise in Convergence, presented at XML Europe 2001 in Berlin. Available from http://www.topicmaps.com/topicmapsrdf.pdf
[9] Lars Marius Garshol - Topic maps, RDF, DAML, OIL. Available from http://www.ontopia.net/topicmaps/materials/tmrdfoildaml.html
[10] Steve Pepper - Ten Theses on Topic Maps and RDF. Available from http://www.ontopia.net/topicmaps/materials/rdf.html
[11] Marc de Graauw 2002 - Survey of Actual Scope Use in Topic Maps. Available from http://www.marcdegraauw.com/files/scope_survey.htm
[12] Marc de Graauw - Structuring Scope. Available from: http://www.marcdegraauw.com/files/structuring_scope.htm
[13] Steve Pepper, Geir Ove Gronmo - Towards a General Theory of Scope. Available from http://www.ontopia.net/topicmaps/materials/scope.htm
[14] Robert Barta, 2003 - Is He The One? Subject Identification in Topic Maps. Available from: http://topicmaps.it.bond.edu.au/docs/21/toc
[15] TMAPI - http://tmapi.org/
[16] Apache Ant - http://ant.apache.org/
[17] Eric Freese - So why aren't Topic Maps ruling the world?, in Extreme Markup Languages 2002: Proceedings. Available: http://www.mulberrytech.com/Extreme/Proceedings/html/2002/Freese01/EML2002Freese01.html
[18] van Rijsbergen, C. J. Information retrieval. Butterworths, 1979.
[19] Salton, Gerard. - Introduction to Modern Information Retrieval. McGraw-Hill, 1983.
[20] Ian H. Witten, Eibe Frank - Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)
[21] Weka - http://www.cs.waikato.ac.nz/ml/
[22] Tapestry - http://jakarta.apache.org/tapestry/
[23] Howard M. Lewis Ship – „Tapestry in Action", Manning Publications Co. (2004), ISBN 1-932394-11-7
[24] FOLDOC http://foldoc.org/
[25] ODP - http://dmoz.org/
[26] Babel - http://www.geocities.com/ikind_babel/babel/babelsr.html
[27] Szószablya - http://mokk.bme.hu/projektek/szoszablya/
[28] YALE - Yet Another Learning Environment http://www-ai.cs.uni-dortmund.de/SOFTWARE/YALE/index.html
[29] jBNC - Bayesian Network Classifier Toolbox http://jbnc.sourceforge.net/
[30] Ka Iok Tong - Enjoying Web Development with Tapestry, ISBN: 1411649133
[31] Jakarta Velocity – http://jakarta.apache.org/velocity/
[32] Ontopia Omnigator – www.ontopia.net/omnigator/
[33] TM4J – http://tm4j.org/
[34] TinyTIM – http://tinytim.sourceforge.net/
[35] TM4WEB – http://tm4j.org/tm4web.html
[36] KEA - http://www.nzdl.org/Kea/
[37] Tolog - http://www.ontopia.net/topicmaps/materials/tolog.html
[38] NIIF Networkshop conference series http://www.iif.hu/rendezvenyek/networkshop/