EU-IST Project IST-2003-506826 SEKT

SEKT: Semantically Enabled Knowledge Technologies

D1.5.1 Extracting human expertise from existing ontologies – software V1.0

Marko Grobelnik, Dunja Mladenic (J.Stefan Institute)

Abstract

Most of the existing ontologies were developed with considerable human efforts. The human expertise captured in the existing ontologies can be extracted to some extent using Knowledge Discovery techniques. This report describes our approach to extracting human expertise from the existing ontologies and the developed software component that is integrated in our software library, TextGarden. We also provide a brief description of TextGarden, a growing software library for handling and analyising text and Web data.

Keyword list: text mining, knowledge discovery, ontology extraction

WP1 Ontology Generation

Prototype PU

Contractual date of delivery: 31.12.2004

Actual date of delivery: 31.12.2004

SEKT Consortium

This document is part of a research project partially funded by the IST Programme of the Commission of the European Communities as project number IST-2003-506826.

British Telecommunications plc.
Orion 5/12, Adastral Park
Ipswich IP5 3RE
UK
Tel: +44 1473 609583, Fax: +44 1473 609832
Contact person: John Davies
E-mail: john.nj.davies@bt.com

Empolis GmbH
Europaallee 10
67657 Kaiserslautern
Germany
Tel: +49 631 303 5540
Fax: +49 631 303 5507
Contact person: Ralph Traphöner
E-mail: ralph.traphoener@empolis.com

Jozef Stefan Institute
Jamova 39
1000 Ljubljana
Slovenia
Tel: +386 1 4773 778, Fax: +386 1 4251 038
Contact person: Marko Grobelnik
E-mail: marko.grobelnik@ijs.si

University of Karlsruhe, Institute AIFB
Englerstr. 28
D-76128 Karlsruhe
Germany
Tel: +49 721 608 6592
Fax: +49 721 608 6580
Contact person: York Sure
E-mail: sure@aifb.uni-karlsruhe.de

University of Sheffield
Department of Computer Science
Regent Court, 211 Portobello St.
Sheffield S1 4DP
UK
Tel: +44 114 222 1891
Fax: +44 114 222 1810
Contact person: Hamish Cunningham
E-mail: hamish@dcs.shef.ac.uk

University of Innsbruck
Institute of Computer Science
Techikerstraße 13
6020 Innsbruck
Austria
Tel: +43 512 507 6475
Fax: +43 512 507 9872
Contact person: Jos de Bruijn
E-mail: jos.de-bruijn@deri.ie

Intelligent Software Components S.A.
Pedro de Valdivia, 10
28006
Madrid
Spain
Tel: +34 913 349 797
Fax: +49 34 913 349 799
Contact person: Richard Benjamins
E-mail: rbenjamins@isoco.com

Kea-pro GmbH
Tal
6464 Springen
Switzerland
Tel: +41 41 879 00
Fax: 41 41 879 00 13
Contact person: Tom Bösser
E-mail: tb@keapro.net

Ontoprise GmbH
Amalienbadstr. 36
76227 Karlsruhe
Germany
Tel: +49 721 50980912
Fax: +49 721 50980911
Contact person: Hans-Peter Schnurr
E-mail: schnurr@ontoprise.de

Sirma AI EAD, Ontotext Lab
135 Tsarigradsko Shose
Sofia 1784
Bulgaria
Tel: +359 2 9768 303, Fax: +359 2 9768 311
Contact person: Atanas Kiryakov
E-mail: naso@sirma.bg

Vrije Universiteit Amsterdam (VUA)
Department of Computer Sciences
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
Tel: +31 20 444 7731, Fax: +31 84 221 4294
Contact person: Frank van Harmelen
E-mail: frank.van.harmelen@cs.vu.nl

Universitat Autonoma de Barcelona
Edifici B, Campus de la UAB
08193 Bellaterra (Cerdanyola del Vall` es)
Barcelona
Spain
Tel: +34 93 581 22 35, Fax: +34 93 581 29 88
Contact person: Pompeu Casanovas Romeu
E-mail: pompeu.casanovas@uab.es

Executive Summary

Text Mining can be defined as a fairly broad research area dealing with computer-supported analysis of text with rather long list of problems that can be addressed. Here we adopt this fairly open view but concentrate on the parts related to knowledge discovery. This report describes our approach to extracting human expertise from the existing ontologies and the developed software component that is integrated in our software library, TextGarden. We also provide a brief description of TextGarden, a growing software library for handling and analyising text and Web data. Some other SEKT deliverables (D1.1.1, D1.2.1, D1.3.1, D1.4.1) describe details of other software components included in the same library.

Most of the existing ontologies were developed with considerable human efforts. The human expertise captured in the existing ontologies can be extracted to some extent using Knowledge Discovery techniques (KDD). We have developed an approach for extracting human expertise from a topic ontology, such as Yahoo or DMoz organizing Web pages according to their content. The approach involves usage of KDD techniques, in particular Text Mining on large collections of Web documents organizaed in topic ontology (Mladenic and Grobelnik 2004). Details on the techniques commonly used to handle dimensionality of the data in topic ontology of documents can be found in (Mladenic and Grobelnik 2003).

Contents

1 Introduction

Text mining is an interdisciplinary area that involves at least the following key research fields:

- Machine Learning and Data Mining (Mitchell 1997; Witten and Frank 1999; Hand, et al. 2001) which provides techniques for data analysis with varying knowledge representations and large amounts of data,
- Statistics and statistical learning (Hastie, et al. 2001) which in the context of text mining contributes data analysis (Duda, et al. 2000) in general,
- Information Retrieval (Rijsberg 1979, Mani and Maybury 1999) providing techniques for text manipulation and retrieval mechanisms, and
- Natural Language Processing (Manning and Schutze 2001) providing the techniques for analyzing natural language. Some aspects of text mining involve the development of models for reasoning about new text documents based on words, phrases, linguistic and grammatical properties of the text, as well as extracting information and knowledge from large amounts of text documents.

As the area is fairly new, there is no publicly available software that can be used as bases for performing research experiments. In over several years of our own research in the area, we have developed different software components that enabled us to perform a number of experiments mainly on real-world data. The software is getting extensions, recently in direction of Semantic Web, some of the part are rewritten as needed and the whole library of components is becoming publicly available.

Here we describe the main parts of the library providing details for the part addressing extraction of human expertise from the existing topic ontologies.

2 Main characteristics of Text Garden

Text Garden is software library with the following technical features.

- The library is written in C++.
- The library compiles under MS VisualC++ (versions 6.0, .NET 2002, .NET2003) and Borland C++ Builder 3.0, 5.0 without any warning. This is achieved by using somewhat conservative C++, while still heavily using templates, some RTTI, and some multiple inheritance. Our plan is to port the library in the near future also to Linux to be compilable with GCC.
- The library currently contains over 100k lines-of code in approximately 200 .h/.cpp files.
- The library is written so that the functionality can be accessed through C++ classes, command-line utilities, ActiveX/COM components, WebService. Currently we provide a publicly available Web page for downloading command-line utilities at http://TexGarden.ijs.si/

Main part of the functionality interesting for SEKT is the set of text-mining classes covering several pipelines (see Figure 1, Figure 2). Basically, the idea is to cover most of the text-mining, information retrieval, web-mining (Chakrabarti 2002), crawling/search scenarios (Figure 2) which one might need in practical applications of basic text-mining techniques. For extracting human expertise from topic ontology addressed in this report, we used the classification part (from Figure 1) and quering Goolge (from Figure 2) as can be seen from the architecture of our approach provided in Figure 3.

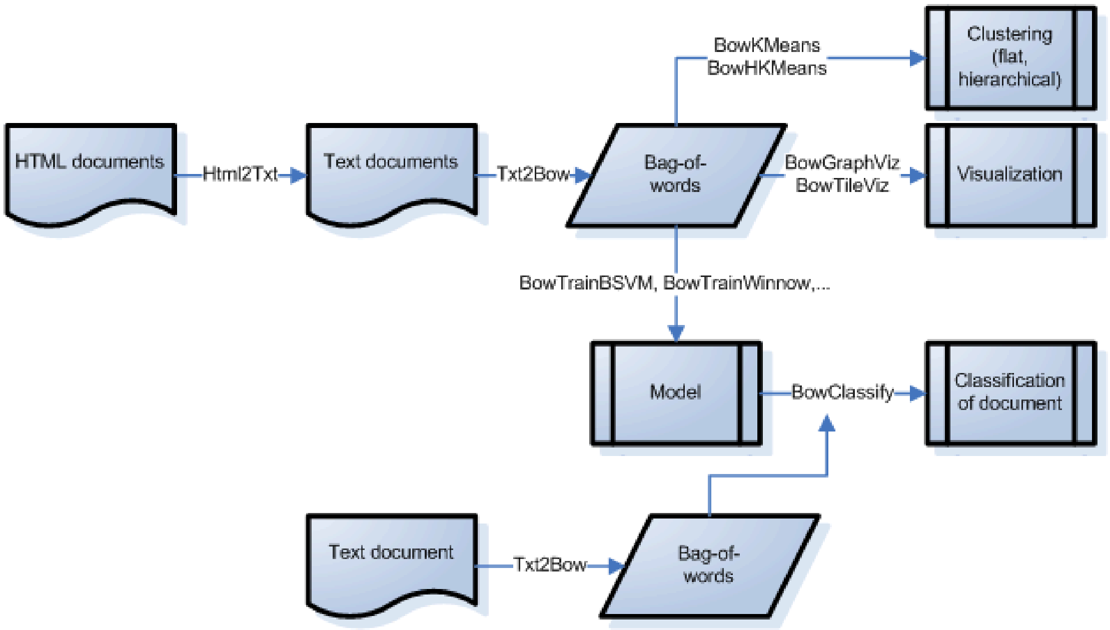


Figure 1 The main scenario with the corresponding components of using the Text Garden library for classification, clustering and visualization of text documents.

3 File formats use for document representation

There are different formats for document representation used in TextGarden covering different ways of handling text documents. For extracting human expertise from topic ontology addressed in this report, we used Bag-Of-Words format. Bag-Of-Words format includes documents in a full processed form supporting the bag-of-words representation. Each document is represented by the set of its word frequencies and categories which it belongs to. The purpose of the document is to perform efficient execution of algorithms working with the bag-of-words representation as: clustering, learning, classification, visualization, etc. There is a utility (Txt2Bow) in TextGarden that we use for transforming text formats into the file in Bag-Of-Words format ".Bow".

4 Functionality of Text Garden

Functionality of the library is split into several groups of utilities as follows. Data preparation including conversions, crawling and similar. Modeling of data including different algorithms for supervised (Sebastiani 2002, Craven and Slattery 2002), unsupervised (Steinbach 2000) and semi-supervised (Nigam et al. 2001) learning (e.g.

SVM, k-means clustering). Efficient implementations of linear-algebra operations being able to deal with large datasets. Visualization of data (Grobelnik and Mladenic 2002) including a couple of methods which are competitive to other state-of the art text visualization (see Figure 1). Indexing and search of textual documents (see Figure 2). Direct support for some popular specific datasets and services such as Google, Amazon, Reuters datasets.

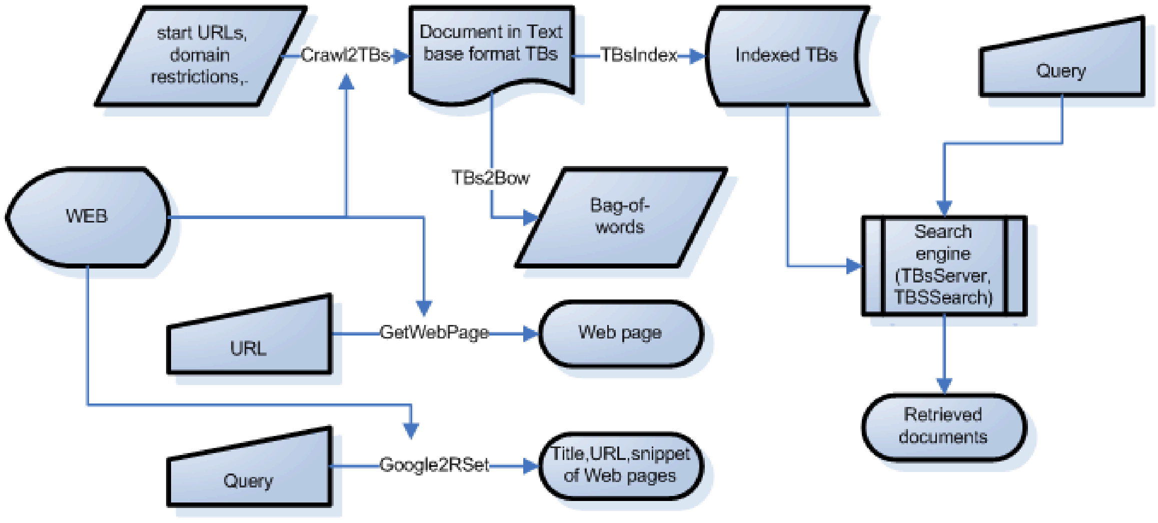

start URLs, domain restrictions,.
Crawl2TBs
Document in Text base format TBs
TBsIndex
Indexed TBs
Query
TBs2Bow
Bag-of-words
WEB
Search engine (TBsServer, TBSSearch)
URL
GetWebPage
Web page
Retrieved documents
Query
Google2RSet
Title,URL,snippet of Web pages

Figure 2 The main scenario with the corresponding components of using the Text Garden library for crawling the Web.

5 Extracting human expertise from existing ontology

Most of the existing ontologies were developed with considerable human efforts. Examples are Yahoo! and DMoz topic ontologies containing Web pages or; MESH ontology of medical terms connected to Medline collection of medical papers. The human expertise captured in the existing ontologies can be extracted to some extent using Knowledge Discovery techniques.

5.1 Related work

Extracting human expertise from existing topic ontology can be defined as a machine learning problem of constructing a model for classifying examples into the existing topic ontology. There is some work in machine learning addressing problems with hierarchically organized features (Shapiro 1987) but problems involving example or classification hierarchy have been only recently explored. An approach to extracting human expertise from a very simple topic ontology was presented in (Koller and Sahami 1997). They used the Reuters news and have manually constructed a topic ontology of them. In that ontology all the documents are placed at the bottom in the leaves representing the most specific topics. Documents are represented as Boolean word-vectors with features representing words selected using greedy algorithm that eliminates features one by one using Cross entropy measure. They compared several learning algorithms and learn document category from the hierarchical structure, dividing classification task into a set of smaller problems corresponding to the splits in the classification hierarchy nodes. They give results on three domains each having a 3-level topic ontology that is based on up to 1,000 documents having 12 nodes.

There is also related work on larger datasets involving existing topic ontology of Web pages. An approach was developed on a part of Yahoo! Ontology of Web pages (McCallum et al., 1998) and the reported are results on extracting expertise from the two bottom layers of a larger ontology. However, there approach does not address the problem of documents being instances of arbitrary rather then just leaf node concepts. Rather they used the same assumption as used in (Koller and Sahami 1997) that all the documents are places in the leaves.

Another approach developed also on the Yahoo topic ontology (Mladenic and Grobelnik 2003, Mladenic and Grobelnik 2004) handles situations when instances are placed in any node of the topic ontology (not just its leaf nodes). Namely, it turned out that some documents were manually placed in the non-leaf nodes as their content is too general for any of the existing leaf nodes (and probably not specific/frequent enough to trigger introduction of a new concept). For a new document, the learned model returns for each topic from the ontology (and the corresponding set of keywords) the probability that the document is an instance of that topic. In that approach, documents are represented as word-vectors and a set of positive and negative examples for each sub-problem are constructed from the given topic ontology. Learning is performed using the naive Bayesian classifier and the final result of learning is a set of specialized classifiers each based only on a small subset of the document vocabulary.

5.2 Description of chosen approach

There are typically two possible approaches to building a hierarchical classifier: (1) flattening of the structure and building separate classifier for each class in the hierarchy/ontology – the final classification is produced from some kind of voting schema, and (2) hierarchical classifier which is appropriate just for taxonomic ontologies where for each node there is a separate classifier deciding which branch from the node should be follow in order to classify a new instance. The solution (1) is more general and allows addressing also non-taxonomic structures, but is computationally more expensive (because in the classification phase it addresses all the classifiers). Solution (2) is more efficient, because it addresses just the number of classifiers which is logarithm of the number of classes in the ontology. Solution (1) is interesting also because it addresses each individual concept in the ontology separately which is not the case in the solution (2) where in the cases of large taxonomic ontologies (such as DMoz) the information about the lower level concepts is lost for the higher level classifiers (which have only very broad view to the distributions about the data in lover branches). We decided to use the solution (1) which proved to be in the combination with kNN algorithm very efficient, because it is able to classify several new documents per second into 600,000 classes (DMoz hierarchy).

We have based our work on our previous work on Yahoo! topic ontology of Web pages (Mladenic and Grobelnik 2004). The same as there, instances in the ontology are html documents, cleaned and represented as word-vectors. We use standard set of parameters for Txt2Bow utility, meaning that we are using standard set of English stop-words (525), Porter stemmer, and generation of phrases as frequent n-grams (n consecutive words) with maximal length of n-grams being 3 and minimal frequency of n-grams being 5. The whole problem is divided into sub problems, one for each concept (topic) of the topic ontology. In order to be able to handle large topic ontologies, such as DMoz (having several million of documents and several hundred thousands concepts), we have used a simple approach to modeling based on k-nearest neighbor algorithm which gives good results in terms of accurate classifications and computational efficiency. The final model of extracted human expertise is provided in the form of a set of specialized classifiers. These classifiers are used when a new document needs to be classified into the topic ontology – we use simple k-nearest voting scheme. An important innovation we used is extending the target document with its context from the web (snippets of the pages pointing to the target document page and, a set of snippets of related documents). In general, it is possible to use other "biased" functions, the problem is only potentially high number of parameters which would need to be trained for such a "biased model". This would be possible for smaller ontologies (several tens of nodes) but is less practical for large ontologies such as DMoz (600,000 classes).

5.3 Architecture

Since we have used the flattened version of DMoz, the approach is similar to classifying into flat set of classes with kNN – the hierarchy is used when flattening the hierarchy where there is several ways how to flatten the structure. Currently we are using the most simple way of flattening – namely, taking all the documents from the sub tree and using them to create a standard centroid vector. In the future we plan to experiment with more sophisticated strategies to create the representative vector of the class – this would include different weighting schemes for sub-trees, using information also from non-taxonomic links.

The architecture of the approach is shown in Figure 3. It consists of the following steps.

1. First, we download the DMoz data from the address http://rdf.dmoz.org/ - the data is structured into two large RDF/XML files – ontology structure (skeleton of the hierarchy) and ontology contents (documents manually indexed into the ontology classes).
2. Since manipulation with large RDF files (approx 2Gb) does not allow for efficient manipulation with the data, we transform the downloaded data with a special utility **DMoz2Bin.Exe** into a binary form – the whole (or part of) DMoz ontology is saved on the file of approx 1Gb. The file represents binary serialized C++ object which has an internal organization allowing querying and traversing the ontology structure and data in very efficient way. E.g. the ontology structure is represented as labeled graph, all the strings are represented in string pool, all the vectors are saved in vector pools etc. which enables efficient storage while preserving manipulability of the data.

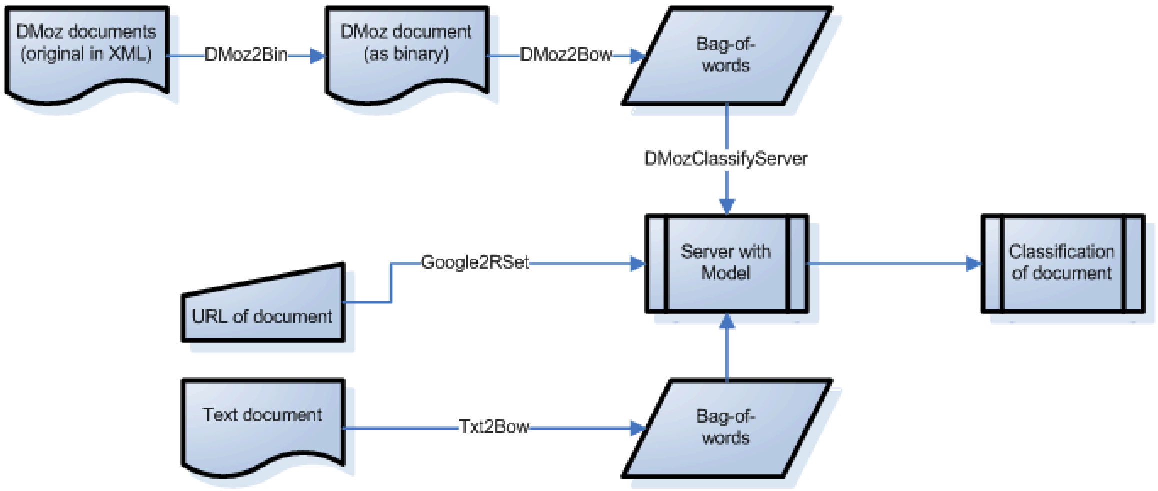


Figure 3 Architecture of the system for extracting human expertise from DMoz topic ontology of Web documents.

3. In the next step, the efficient binary structure from the previous step is used to create a set of TFIDF bag-of-word vectors which are used as a basis for the k-nearest neighbor classification algorithm. This is performed with the special utility **DMoz2Bow.Exe** which creates two files from the ontology documents: ".Bow" file with bag-of-words vectors and ".BowPart" with the mapping of the vectors into the structure. In this representation we calculate for each node in the topic ontology a centroids vector of all the documents (short documents describing indexed web pages) in the node itself and its sub-tree. In other words – each bag-of-words unit represents a union of all the documents belonging to the concept.
4. The data prepared in the pervious steps is used for classification. The classification model consist of a set of vectors, as already proposed in (Mladenic and Grobelnik 2003) each representing a single node from the topic ontology. Classification of a new document into topic ontology consists from finding the concepts whose centroids vector is the most similar to the target document. For calculating similarity between the document and the concept centroids vector we use standard cosine measures on TFIDF vectors.
5. If the target document which we want to classify has also a URL address, we enrich the document with its context consisting from two parts: snippets of the pages pointing to our target document (using link Google function) and snippets of the pages which are related to our target page (Google target page).
6. For classification we have developed DMoz classification server (**DMozClassifyServer.Exe**) which loads the model data into the memory enabling for efficient k-nearest neighbor classification. The software offers functionality as web user interface or as a web service (providing results in XML format). On the input, we provide URL (if available) of the target document and the text of the document. On the output, we get a list of the most probable categories (concepts) from DMoz topic ontology with associated weights and a list of the most probable keywords (calculated from the path segments from the names of the DMoz categories).

5.4 Future development

In the future development we plan to extensively evaluate the approach for classification of new documents into the topic ontology. The goal is to reconstruct human skills when manually classifying the documents into ontology as best as possible. In addition to DMoz topic ontology, we plan to repeat the same procedure for several other datasets such as Medline (large medical ontology), and the SEKT case studies as needed.

6 Appendix – User Guide

DMoz2Bin

DMoz2Bin converts the data downloaded from http://rdf.dmoz.org/ into an efficient binary representation which could be used in further steps of dealing with DMoz data. The files from DMoz web site needed for processing are "structure.rdf.u8" and "content.rdf.u8". On the output the utility produces the file "DMozFull.DMoz" which includes complete information from DMoz (structure and contents).

```
usage: DMOZ2BIN.EXE
   -i:Input-File-Path (default:") – input path
   -o:Output-File-Path (default:'') – output path
   -struct:Structure-Only (default:'F') – process just the structure (ignore the contents)
   -m: Memory-Pool-Size (default:830000000) – size of the memory pool
```

DMoz2Bow

DMoz2Bow extracts from ".DMoz" file (prepared with DMoz2Bin) the whole taxonomy (or part of it) into ".Bow" file (with TFIDF vectors of the documents) and ".BowPart" file (having the mapping between the documents and the tree).

```
usage: DMOZ2BOW.EXE
   -i:Input-DMoz-File-Path (default:") – input path with ".DMoz" file
   -o:Output-File-Path (default:") – output path where the output files are generated
   -c:Root-Category-Name (default:'Top') – name of the root-node
   -sc:Sub-Categories (default:) – subcategories of the root node (if empty, take all)
   -stopword:Stop-Word-Set (en8, en425, en523, ge) (default:'en523') – stop words
   -stemmer:Stemmer (Porter) (default:'porter') – stemming
   -ngramlen:Max-NGram-Length (default:3) – maximal length of word n-grams
   -ngramfq:Min-NGram-Frequency (default:5) – minimal frequency of n-grams
```

DMozClassifyServer

DMozClassifyServer takes on the input ".Bow" and ".BowPart" files and generates a web server application which on a request classifies a new document into DMoz categories.

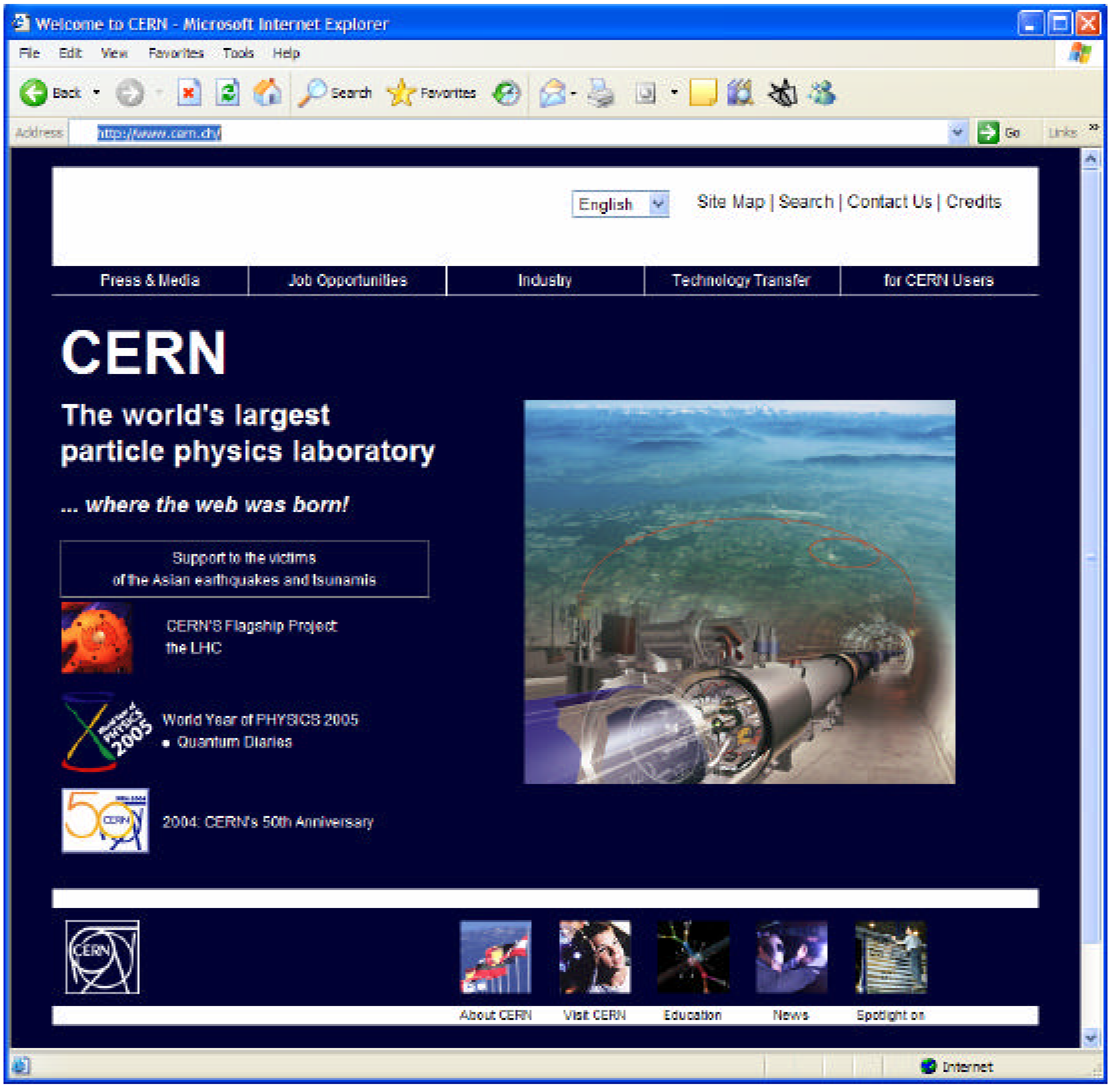
```
usage: DMOZCLASSIFYSERVER.EXE
   -ibow:Input-BagOfWords-FileName (default:'') – “.Bow” filename
   -ipart:Input-BagOfWords-Partition-FileName (default:'') – “.BowPart” filename
   -ot:Output-Logging-Txt-File (default:'DMozClassifyLog.Txt') – logging in text file
   -ox:Output-Logging-Xml-File (default:'DMozClassifyLog.Xml') – logging in xml file
   -port:Server-Port (default:8888) – web-server port number
```

Example:
We run the server on the Science part of DMoz with the following call:

```
> DMozClassifyServer.exe -ibow:f:\Data\DMoz\Top_Science.Bow
-ipart:f:\Data\DMoz\Top_Science.BowPart

Classification into DMoz - Server
========================
Input-BagOfWords-FileName (-ibow:)=f:\Data\DMoz\Top_Science.Bow
Input-BagOfWords-Partition-FileName (-ipart:)=f:\Data\DMoz\Top_Science.BowPart
Output-Logging-Txt-File (-ot:)=DMozClassifyLog.Txt
Output-Logging-Xml-File (-ox:)=DMozClassifyLog.Xml
Server-Port (-port:)=8888
========================
Web-Server: Started at port 8888.
Loading bag-of-words data from 'f:\Data\DMoz\Top_Science.Bow' ... Done.
Loading bag-of-words-partition from 'f:\Data\DMoz\Top_Science.BowPart' ... Done.
```

Next, we want to classify into Science part of DMoz the following page (CERN institute home page):



CERN

The world's largest particle physics laboratory

... where the web was born!

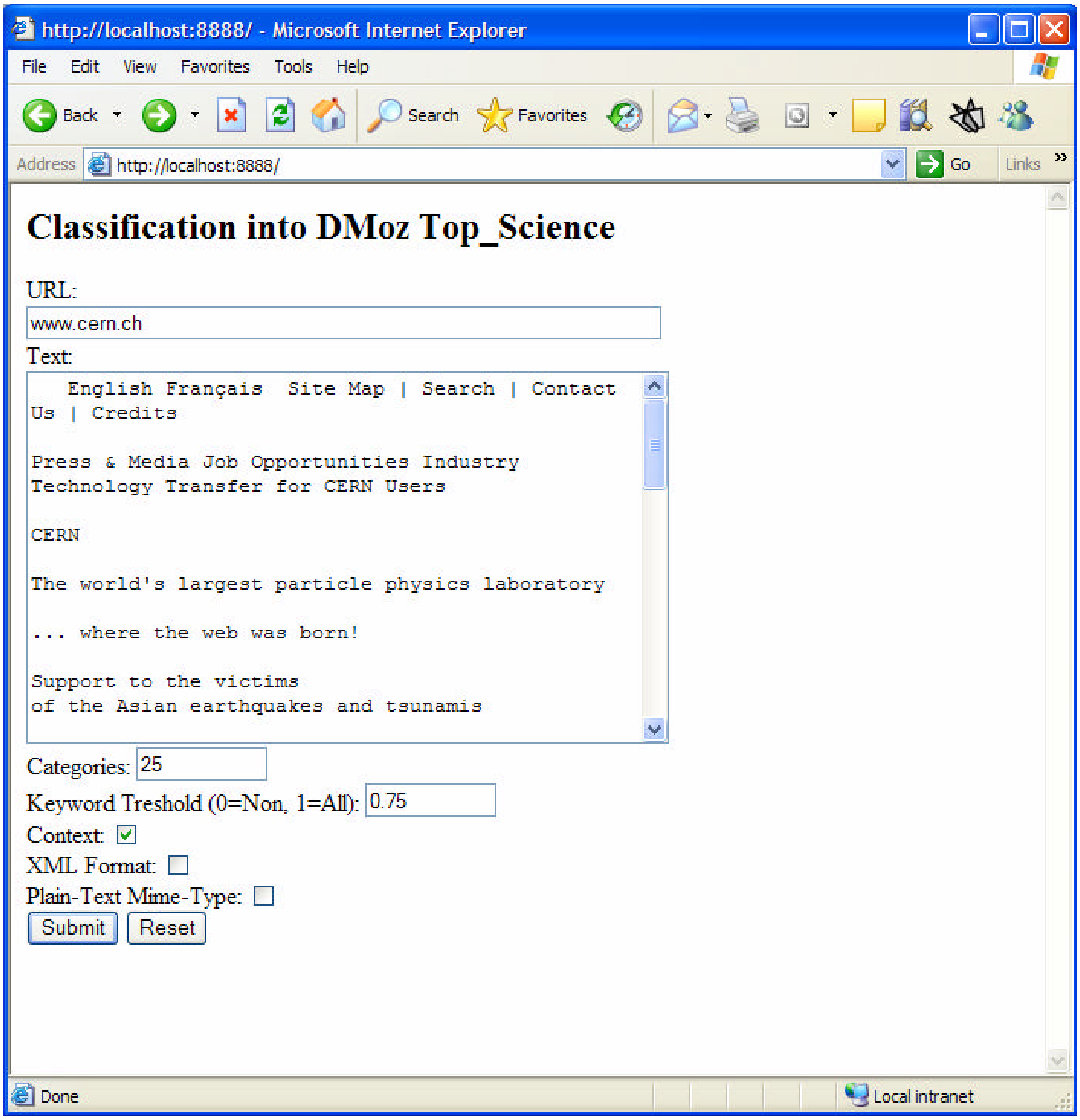
Support to the victims of the Asian earthquakes and tsunamis

CERN'S Flagship Project the LHC

World Year of PHYSICS 2005
- Quantum Diaries

2004: CERN's 50th Anniversary

In the web browser we open the page offered by DMozClassifyServer (we use localhost:8888 address since in this example we use local machine for server and client). We type in the URL of the page and the text from the page. Classification works also in the case when one of the information (either URL or text) is missing.


http://localhost:8888/ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://localhost:8888/
Go
Links
Done
Local intranet

Classification into DMoz Top_Science

URL: www.cern.ch

Text:

```
   English Français  Site Map | Search | Contact
Us | Credits

Press & Media Job Opportunities Industry
Technology Transfer for CERN Users

CERN

The world's largest particle physics laboratory

... where the web was born!

Support to the victims
of the Asian earthquakes and tsunamis
```

Categories: 25

Keyword Treshold (0=Non, 1=All): 0.75

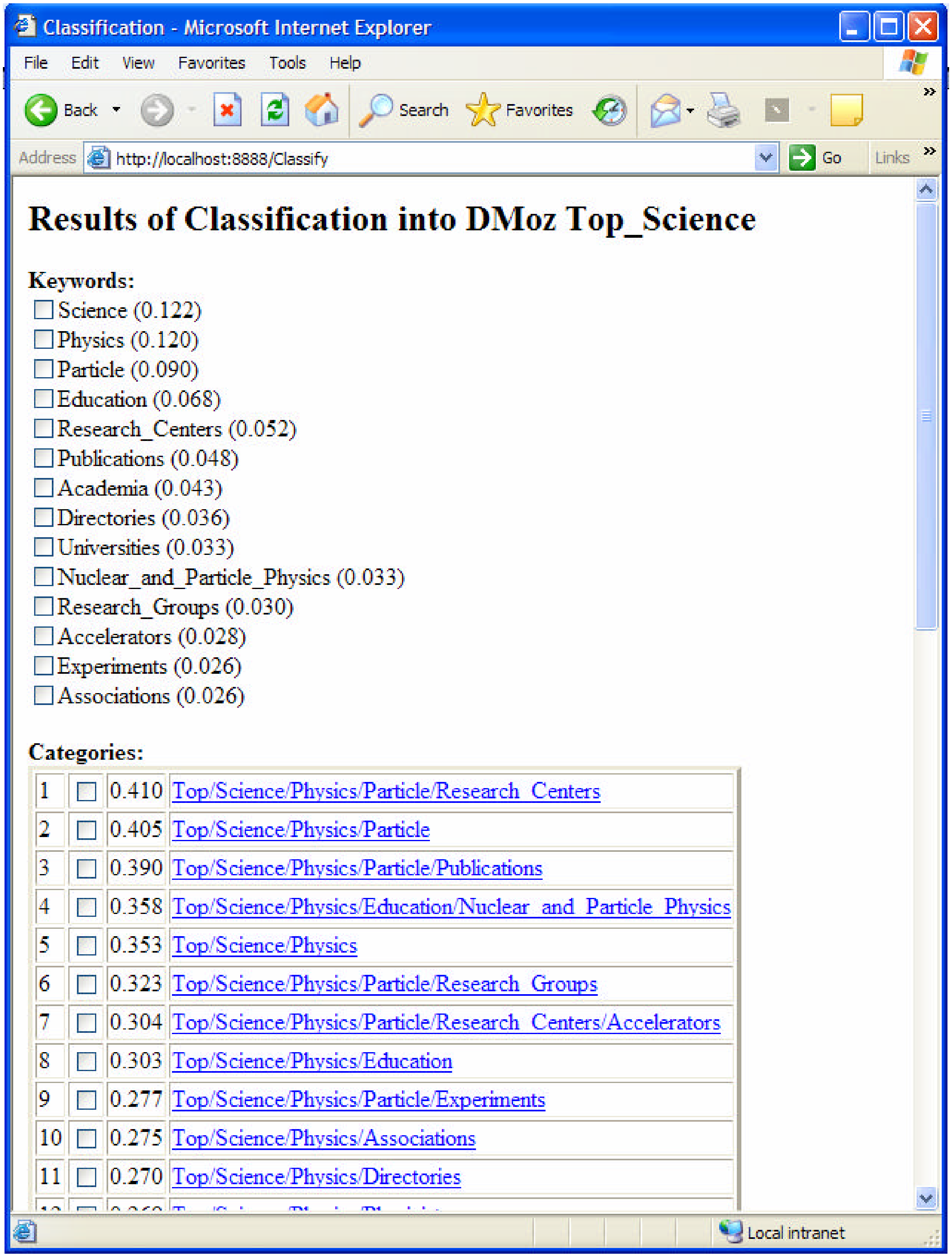
Context: ☑

XML Format: □

Plain-Text Mime-Type: □

Submit Reset

After pressing “Submit” button the server sends back the list of keywords which are the most relevant for the submitted page and the list of most relevant categories from DMoz:



Results of Classification into DMoz Top_Science

Keywords:

☐ Science (0.122)
☐ Physics (0.120)
☐ Particle (0.090)
☐ Education (0.068)
☐ Research_Centers (0.052)
☐ Publications (0.048)
☐ Academia (0.043)
☐ Directories (0.036)
☐ Universities (0.033)
☐ Nuclear_and_Particle_Physics (0.033)
☐ Research_Groups (0.030)
☐ Accelerators (0.028)
☐ Experiments (0.026)
☐ Associations (0.026)

Categories:

1	☐	0.410	Top/Science/Physics/Particle/Research_Centers
2	☐	0.405	Top/Science/Physics/Particle
3	☐	0.390	Top/Science/Physics/Particle/Publications
4	☐	0.358	Top/Science/Physics/Education/Nuclear_and_Particle_Physics
5	☐	0.353	Top/Science/Physics
6	☐	0.323	Top/Science/Physics/Particle/Research_Groups
7	☐	0.304	Top/Science/Physics/Particle/Research_Centers/Accelerators
8	☐	0.303	Top/Science/Physics/Education
9	☐	0.277	Top/Science/Physics/Particle/Experiments
10	☐	0.275	Top/Science/Physics/Associations
11	☐	0.270	Top/Science/Physics/Directories

References

1. Chakrabarti. S., (2002). Mining the Web: Analysis of Hypertext and Semi Structured Data, Morgan Kaufmann.
2. Duda, R. O., Hart, P. E. and Stork, D. G. (2000). Pattern Classification 2nd edition, Wiley-Interscience.
3. Fayyad, U., Grinstein, G. G. and Wierse, A. (editors), (2001). Information Visualization in Data Mining and Knowledge Discovery, Morgan Kaufmann.
4. Grobelnik, M., and Mladenic, D., (2002). Efficient visualization of large text corpora. Proceedings of the seventh TELRI seminar. Dubrovnik, Croatia.
5. Hand, D.J., Mannila, H., Smyth, P. (2001) Principles of Data Mining (Adaptive Computation and Machine Learning), MIT Press.
6. Hastie, T., Tibshirani, R. and Friedman, J. H. (2001). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Series in Statistics, Springer Verlag.
7. Jackson, P., Moulinier, I., (2002). Natural Language Processing for Online Applications: Text Retrieval, Extraction, and Categorization, John Benjamins Publishing Co.
8. Koller, D., Sahami, M., (1997). Hierarchically classifying documents using very few words, Proceedings of the 14th International Conference on Machine Learning ICML-97, pp. 170-178, Morgan Kaufmann, San Francisco, CA.
9. Nigam, K., McCallum, A., Thrun, S., and Mitchell, T., (2001). Text Classification from Labeled and Unlabeled Documents using EM, Machine Learning Journal.
10. McCallum A., Rosenfeld R., Mitchell T., Ng A., (1998). Improving Text Classification by Shrinkage in a Hierarchy of Classes, Proceedings of the 15th International Conference on Machine Learning ICML-98, Morgan Kaufmann, San Francisco, CA.
11. Mani, I., Maybury, M.T. (editors), (1999). Advances In Automatic Text Summarization, MIT Press.
12. Manning, C.D., Schutze, H. Foundations of Statistical Natural Language Processing, The MIT Press, Cambridge, MA, 2001.
13. Mitchell, T.M. (1997). Machine Learning. The McGraw-Hill Companies, Inc.
14. Mladenic, D. (2002). Web browsing using machine learning on text data, In (ed. Szczepaniak, P. S.), Intelligent exploration of the web, 111, Physica-Verlag, 288–303.
15. Mladenic, D., Grobelnik, M. (2003). Feature selection on hierarchy of web documents. Journal of Decision support systems, 35, 45-87.
16. Mladenic, D., Grobelnik, M. (2004). Mapping documents onto web page ontology. In: *Web mining : from web to semantic web* (Berendt, B., Hotho, A., Mladenic, D., Someren, M.W. Van, Spiliopoulou, M., Stumme, G., eds.), Lecture notes in artificial inteligence, Lecture notes in computer science, vol. 3209, Berlin; Heidelberg; New York: Springer, 2004, 77-96.
17. Rijsberg, C. J., van (1979), Information Retrieval, Butterworths.
18. Craven, M., Slattery, S., (2001). Relational learning with statistical predicate invention: Better models for hypertext. Machine Learning, 43(1/2):97-119.

19. Sebastiani, F., Machine Learning for Automated Text Categorization, ACM Computing Surveys, 2002.
20. Steinbach, M., Karypis, G. and Kumar, V. (2000). A comparison of document clustering techniques. Proc. KDD Workshop on Text Mining. (eds. Grobelnik, M., Mladenic, D. and Milic-Frayling, N.), Boston, MA, USA, 109–110.
21. Witten, I.H., Frank, E., (1999) Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann.