13th European Conference on Machine Learning (ECML'02)
6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'02)

2nd Workshop
Semantic Web Mining

Bettina Berendt, Andreas Hotho, Gerd Stumme

August 20, 2002
Helsinki
Finland

Foreword

The Semantic Web and Web Mining are two fast-developing research areas, which have many points in contact. A growing number of Web resources are semantically annotated, usually in XML or RDF. Data mining algorithms are becoming standard modules in data analysis packages, and their application is becoming a basis for Web reporting and controlling measures. However, integrated theories and software solutions are still lacking. The workshop aims to advance the convergence between Semantic Web and Web mining research by bringing together researchers and practitioners from these two areas. Our aim is to improve, on the one hand, the results of Web mining by exploiting the new semantic structures in the Web, and on the other hand to exploit Web mining for building the Semantic Web.

Web mining applies data mining techniques on Web content, usage, and structure. Methods of Web content mining can, on the one hand, be used to *create* semantic annotations from Web page content; on the other hand, content mining can profit from content that is already *structured* in XML, RDF, or ontological format. Methods of Web usage mining can profit from semantically enriched descriptions of the Web pages visited; this will provide for the identification of more meaningful patterns within site visits, and better site improvements, recommendations, and personalization options based on these patterns. Usage patterns can in turn serve to improve the semantic annotations of pages. The third form of Web Mining, Web structure mining, utilizes the hyperlink structure. Crawlers that take into account structure as well as semantic content can significantly improve search engine results. The rapid development of the field means that Semantic Web Mining now plays a wide range of different roles.

Our workshop aims to bring together researchers and practitioners from the involved fields, and to foster the exchange of ideas, the discussion of currently proposed solutions, and the establishment of an agenda for further emerging issues. In the contributions to this workshop, four categories can be distinguished:

Two invited papers give an overview on different uses of data mining techniques within Semantic Web Mining: *Claire Nédellec* discusses the use of Machine Learning and Information Extraction for annotating text, with the aim to build the Semantic Web. The approach is illustrated by an example taken from genomics. *Nada Lavrač* discusses Inductive Logic Programming techniques for learning from the Semantic Web. She focusses especially on Relational Data Mining and Subgroup Discovery.

The invited talks are complemented by three full papers which focus on basic research: *Andreas Faatz* and *Ralf Steinmetz* explain how an existing ontology can be enriched by mining the web. The authors gather a text corpus using the Google search engine, and compare it with the concepts of the ontology based on statistical information on word usage. *Honghua Dai* and *Bamshad Mobasher* discuss a Semantic Web Usage Mining approach. They present a way of using domain ontologies to automatically characterize usage profiles. Multi-faceted learning for building web taxonomies is the topic of the paper by *Wray Buntine* and *Henry Tirri*. They present results for multi-faceted clustering of bigram words.

Two application papers discuss the use of existing Data/Web mining methods to Semantic Web related problems/data: *Chung-Hong Lee* and *Hsin-Chang Yang* consider multilingual text corpora and cluster them using self-organizing maps. The documents are re-categorized based on their semantic relatedness in the corpus. *Peter Edwards*, *Gunnar AAstrand Grimmes*, and *Alun Preece* empirically investigate the hypothesis that learning from the Semantic Web will outperform traditional learning from today's Web. They compare the performance and accuracy of ILP against k-Nearest Neighbor and Naïve Bayes on two datasets.

Lastly, an important aim of this workshop is to promote the exchange of ideas for the benefit of ongoing research efforts. This is reflected in four project descriptions.

With this collection of research papers, we aim to further the convergence of the Semantic Web and Web mining. We wish to express our appreciation to the authors and to the members of the program committee, for making the workshop a valuable contribution to Semantic Web Mining.

Last but not least we wish to thank KDNet for the support provided for the workshop. Our aim is to use this workshop as a first discussion forum for further activities within KDNet. We want to clarify and discuss further interests and needs concerning Semantic-Web-related activities within KDNet. These could include the organization of further workshops, as well as the participation in project initiatives, mailing lists and community fora. Of major interest in this direction is the initiative of the Web Mining Forum as planned by KDNet.

July 2002

Bettina Berendt
Andreas Hotho
Gerd Stumme

Organization

The Semantic Web Mining Workshop was organized as a workshop within the 13th European Conference on Machine Learning (ECML'02) and the 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'02). It was held on August 20, 2002, in Helsinki, Finland.

Workshop Chairs

Bettina Berendt
Institut für Wirtschaftsinformatik
Humboldt-Universität zu Berlin
Spandauer Str. 1
D–10178 Berlin, Germany
http://www.wiwi.hu-berlin.de/˜berendt
berendt@wiwi.hu-berlin.de

Andreas Hotho
Institut für Angewandte Informatik
und Formale Beschreibungsverfahren (AIFB)
Universität Karlsruhe
D–76128 Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de/WBS/aho
hotho@aifb.uni-karlsruhe.de

Gerd Stumme
Institut für Angewandte Informatik
und Formale Beschreibungsverfahren (AIFB)
Universität Karlsruhe
D–76128 Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de/WBS/gst
stumme@aifb.uni-karlsruhe.de

Program Committee

Stefan Decker
(Stanford University, Palo Alto)

Ronen Feldman
(Bar-Ilan University, Ramat Gan)

Stefan Haustein
(Universität Dortmund)

Jörg-Uwe Kietz
(kd labs ag, Zürich)

Nada Lavrač
(Jozef StefanInstitute, Ljubljana)

Klaus-Peter Huber
(SAS, Heidelberg)

Alexander Mädche
(FZI Research Center for Information Technologies at the University of Karlsruhe)

Bamshad Mobasher
(DePaul University, Chicago)

Claire Nédellec
(Université Paris Sud)

John R. Punin
(Rensselaer Polytechnic Institute, Troy, N.Y.)

Myra Spiliopoulou
(Handelshochschule Leipzig)

Rudi Studer
(Universität Karlsruhe)

Stefan Wrobel
(Rheinische Friedrich-Wilhelms-Universität Bonn)

Table of Contents

Invited Talks

Research Papers

Application Papers

Project Descriptions

Machine Learning applied to Information Extraction in specific domains — an example, gene interaction extraction from bibliography in genomics

Claire Nédellec

Laboratoire Mathématique, Informatique et Génome (MIG), INRA

nedellec@versailles.inra.fr

1. Introduction

As well as the generalization of multimedia communication, the volume of textual information is exponentially increasing. Today mere Information Retrieval technologies are unable to meet specific information needs because they provide information at a document collection level. Developing intelligent tools and methods, which can give access to document content, is therefore more than ever a key issue for knowledge and information management. Text content access is a crucial issue as much in the document engineering system of a small firm as in the document management of a whole scientific domain, whichever the source of information is: an Intranet information system or the "semantic web".

As soon as one wants to automate access to the content of texts in electronic form, one needs semantic knowledge to localize and interpret the relevant information. The acquisition of semantic knowledge is a well-known bottleneck for real-world applications, whichever technology is used (Information Extraction, Question/Answering, and more generally document engineering). There are two main reasons. Firstly, little semantic knowledge specific to application domains has been available because, until now, effort has been mainly devoted to the definition of formal languages for the representation of ontology and to the acquisition of generic knowledge bases, either lexical databases such as WordNet or EuroWordNet or general ontologies; CYC, for instance. In contrast, almost no community effort has been devoted to the acquisition of specific semantic knowledge that is required for particular applications and to the design of the acquisition methods that could be applied. We claim that no generic knowledge can be used as such and that the required semantic knowledge, even if it is derived from a generic source, must be specifically tuned to the application, domain and task that it will be used for. Although the process of acquiring this specific semantic knowledge cannot be fully automatic, methods and tools can be designed to efficiently help its acquisition. Secondly, it is also noticeable that there has been little dialogue between the various disciplines involved in knowledge acquisition and text analysis, although the integration of methods and tools from various disciplines is obviously needed. These disciplines include Information Science, Linguistics, Natural Language Processing, Knowledge Acquisition, Knowledge Representation, Machine Learning, Information Retrieval and Information Extraction. The Caderige project (http://www-caderige.imag.fr) is an example of such a collaboration in the domain of functional genomics. It involves four French laboratories, IRISA (ML and NLP), LIPN (KA, KR and NLP), LRI (ML) and MIG (genomics, ML and IE) and more recently a biotechnology company, Hybrygenics.

After sequencing, the next challenge in genomics is identifying the role of genes in interaction networks. Genome research projects have resulted in new experimental approaches, such as using DNA chips, at the level of whole organisms. Such chips provide comprehensive data about gene activity, so a research team can quickly produce thousands of measurements. More than ever, these new lab technologies are calling for fast and efficient access to previous results to interpret elementary measurements from the laboratory. Unfortunately, most functional genomics knowledge is not described in databanks; it is only available in scientific abstracts and articles written in natural language. For instance, the main generalist bibliographic database, Medline, contains approximately 12 millions entries. Efficiently using previous research results requires automating access to bibliography content. Therefore, exploring bibliographies and extracting knowledge from literature is a major milestone toward developing functional models of gene interactions.

In our opinion this new challenge offers as many benefits and present the same level of technical difficulty as other more popular bioinformatics challenges such as designing predictive algorithmic models. Moreover, AI research in natural language processing (NLP), information extraction (IE), machine learning (ML), and genomics have now reached the stage where automating IE from genomics literature is a realistic and exciting research goal. The specificity of the genomics bibliography, compared to other domains, justifies the expectation for short-term and high-quality results. We will illustrate this claim in the following by a genomic example about information extraction of gene interaction in *Bacillus subtilis*.

2. An example of the IE problem in genomics

Biologists can search bibliographic databases via the Internet using keyword queries that retrieve a large superset of relevant papers. Alternatively, they can navigate through hyperlinks between genome databanks and the corresponding papers. To extract the requisite gene interaction knowledge from the retrieved papers, they must identify the relevant fragments (see the bold text in Figure 1). Such manual processing is time consuming and repetitive, because of the bibliography size, the relevant data sparseness, and the database continuous updating.

UI - 99175219

AB - GerE is a transcription factor produced in the mother cell compartment of sporulating Bacillus subtilis. **It is a critical regulator of cot genes encoding proteins that form the spore coat late in development.** Most cot genes, and the gerE gene, are transcribed by sigmaK RNA polymerase. **Previously, it was shown that the GerE protein inhibits transcription in vitro of the sigK gene encoding sigmaK.** Here, we show that GerE binds near the sigK transcriptional start site, to act as a repressor […]

Figure 1. An extract of a Medline abstract on transcription in Bacillus subtilis.

Interaction	**Type**: negative **Agent**: GerE protein		
	Target:	**Expression**	**Source**: sigK gene **Product**: sigmaK protein

Figure 2. Information extracted from the second selected fragment

For example, the query “Bacillus subtilis and transcription” retrieves 2,209 abstracts such as the one of Figure 1. We chose this query example because *Bacillus subtilis* is a model bacterium and *transcription* is a central phenomenon in functional genomics. Gene functions are realized through gene transcription and protein production. The example of Figure 1 represents the problems posed by applying IE to a bibliography in genomics. Extraction involves understanding and requires expertise in biology. The information to be extracted is sparse in the document set. For instance, in the set of 2,209 abstracts I mentioned, only 3 percent of the sentences contain relevant information on gene interaction—that is, text that mentions the interaction’s agents and type. Hopefully, in biology the bibliography is well structured and the information is local, mainly located in a single sentence or in a part of it as opposed to other domains where it is spread over the document. Many other biological phenomenon, such as translation or gene homology, raise similar IE problems.

3. Limitations of usual IE methods

Up to now, DARPA’s MUC (Message Understanding Conference) program has defined automatic IE as the task of extracting specific, well-defined types of information from natural language texts in restricted domains. The objective is to fill predefined template slots and databases, such as shown in Figure 1b. In functional genomics, even such a restrictive view of IE is useful. Until now, no operational

IE tool has been made available in genomics, and extraction has not been automated.

However, applying IE à la MUC to genomics and more generally to biology is not an easy task because deep text analysis methods are needed to handle the relevant fragments. IE systems should combine the semantic-conceptual analysis of text understanding methods with IE through pattern matching, [Thomas *et al.*, 2000], [Blaschke *et al.*, 99], [Sekimizu *et al.*, 98], [Ono *et al.*, 2001]. Indeed, IE approaches to genomics, based either on predefined sets of fixed patterns, or on shallow representations of the text, yield limited results with either a bad recall or a low precision.

Hand-coded sets of patterns based on significant interaction verbs, gene names, or even syntactic tags and dependencies, [Blaschke *et al.*, 99], [Thomas *et al.*, 2000], [Ono *et al.*, 2001], retrieve little high-quality information. Our experiments with such patterns (described in the IE literature in genomics)—for example, [(Protein1/Gene1) *[1] (interact/associate/bind) * (Protein2/Gene2) *]—yield a precision around 98 percent with a recall between 0 and 20 percent. The reason is that, even in technical and scientific domains, there are many ways to express given biological knowledge in natural language. Manually encoding all patterns encountered in a corpus is thus unfeasible due to cost and unreliability. Therefore, automatically learning such IE patterns or rules from corpus seems to be an appropriate solution. Additionally, building IE systems is time consuming if they rely on manually encoded dictionaries and extraction rules or patterns that are specific to the domains and tasks at hand and they are not easily portable.

At the opposite end, some methods are based on statistic measures of keywords and gene name co-occurrences, [Craven, 99] (for example, shallow information-retrieval-based techniques), [Blaschke *et al.*, 99]. They yield high recall and low precision because they assume that any pair of genes encountered in the retrieved sentences interact, which is not always true. Many false positives are thus retrieved because potentially discriminant keywords and gene names occur in sentences where the genes mentioned are *not* semantically related. The following example (Figure 3.a and 3.b) illustrates some of the problems encountered by both hand-coded patterns and statistic –based approaches. Figure 3.a gives an example of a sentence that cannot be handled by these approaches and Figure 3.b represents the correct gene interaction network that should be extracted from this sentence.

"**GerE** stimulates **cotD** transcription and inhibits **cotA** transcription in vitro by **sigma K** RNA polymerase, as expected from in vivo studies, and, unexpectedly, profoundly inhibits in vitro transcription of the gene (**sigK**) that encode **sigma K**.". The sentence describes five interactions, sigma K with **cotA** and **cotD**, GerE with **cotD**, with **cotA** and with sigK.

Figure 3.a An example of sentence that cannot be handled by hand-coded patterns and pure statistic–based approaches

An intuitive pattern, such as the one mentioned above, (i. e. [(Protein1/Gene1) * (interact/associate/bind) * (Protein2/Gene2) *]), that would match any pair of gene or protein names and interaction verbs or nouns (framed in the figure 3.a), [Craven & Kumlien, 1999], [Blaschke *et al.*, 99], would retrieve many erroneous interactions from this sentence, such as cotD [...] inhibits [...] cotA. Additional criterion such as a maximum number of words between gene names would yield a better precision but would miss some interactions such as the inhibition of sigK gene transcription by GerE (28 words apart). Statistics and keyword-based approaches would select the relevant sentences but would not be able to determine the right interactions between the five different gene and protein names cited in Figure 3.a (in bold-faced text).

[1] * matches any string of any length (including zero).

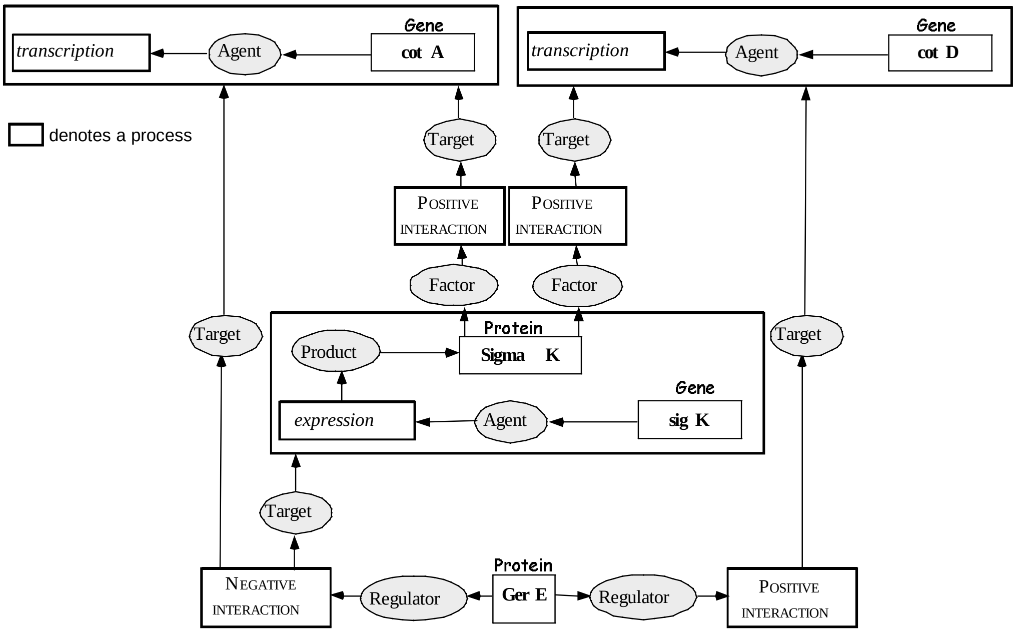


Figure 3.b The gene interaction network to be extracted.

Extracting relevant knowledge in the selected documents thus requires deeper syntactic and semantic analysis based on lexical and semantic resources specific to the domain. For instance, in Figure 2.a, identifying that GerE is the subject of the verb “inhibit” and that sigK is its direct object, and given that these relations are compatible with the conceptual agent and target roles, would improve the extraction’s quality. To summarize, extraction patterns should be learned, because their manual development is unfeasible, and the learning should be based on syntactic–semantic regular expressions.

4. ML and IE today

Since the beginning of the nineties, automatically learning extraction rules from examples of pairs of filled patterns and annotated documents seemed like an attractive approach.[6] However, by the end of the decade, people were questioning the relative merits of the trainable and the knowledge engineering approaches—Doug E Appelt and David J. Israel, for example, discussed this issue at an IJCAI-99 (Int’l Join Conference on Artificial Intelligence) tutorial on IE (http://www.ai.sri.com/~appelt/ie-tutorial/). According to them, trainable (that is, statistics and ML-based) approaches should be preferred when the training data is cheap and plentiful, the extraction specifications are stable, and obtaining the highest possible performance is not a critical issue. They consider that the best recall the ML-based systems obtained is quite low compared to hand-coded IE systems. Appelt and Israel’s analysis is based on the current state of the art in IE, in which existing ML-based systems exploit little, if any, background knowledge for guiding learning. The systems are often applied to a rather shallow representation of the training texts, and most of them are based on general-purpose ML algorithms —mainly *K* nearest-neighbor, grammatical inference, naïve Bayes methods, and top-down or bottom-up relational learning based on an exhaustive search or a local information gain measure.

Two related facts explain the limited range of these approaches, despite the rich spectrum of the modern state of the art in ML. First, according to the limited experiments performed, [Freitag, 98], on the common and quite simple IE tasks (MUC tasks, IE on the job, and seminar announcements), approaches based on linguistic analysis, lexical semantics, and informative representation of the training data do not perform much better than more shallow approaches. This does not encourage the design and application of novel symbolic and relational ML methods, which would be suitable for richer text analysis. Second, until recently, the main stream in text processing was mainly linguistic and statistic but not ML-based, besides some notable exceptions such as S. Soderland's work and T. Mitchell's group research, [Soderland, 99] [Freitag, 98]. A large part of the effort in learning for IE, including genomic applications, has also been devoted to lower-level tasks such as named entity recognition, [Fukuda, 98]. This situation is evolving with the growing interest of the ML community in text processing and in IE in particular. Moreover, the growing demand for applications brings many new IE tasks, such as IE in functional genomics, that require a deeper understanding and consequently call for more sophisticated linguistics- and ML-based approaches [Craven & Kumlien, 99]. Additionally, in real-world applications, training complements rather than opposes knowledge engineering, as ontology-based and interactive approaches illustrate.

5. Linguistics- and ML-based approach of IE in future genomics

In Caderige, we view the genomics IE of the future as a three-step method. In the first step, we select the relevant textual fragments from all sentences in the papers, based on shallow criteria (for example, discriminant keywords or gene and protein names) to deal with the relevant data's sparseness. In the second step, we build a representation of the content of the fragments using successive interpretation operations based on syntactic–semantic lexicon, [Sekimizu *et al.*, 99], [Rindflesh *et al.*, 2000], following a classical approach in text understanding. Figure 4 shows an example of this phase's output. This step should involve terminology, ontologies, and predicate argument structures to label the relevant terms and syntactic dependencies with the appropriate concepts. In doing so, we rely on the fact that in the language of a given specific domain there exists strong syntactic regularities, which make it possible to build a semantic structure.

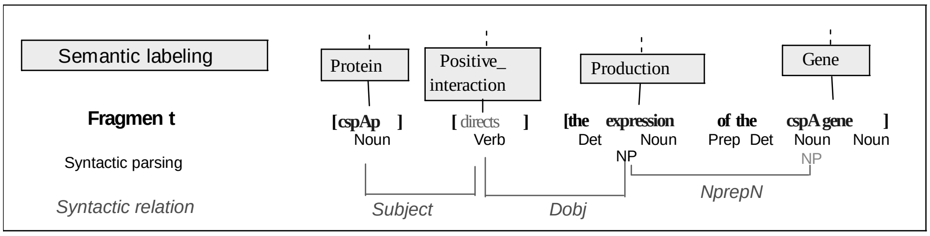


Figure 4. Example of syntactic-semantic interpretation (NP denotes noun phrases and Dobj denotes the Direct Object).

Finally, we apply extraction rules (see Figure 5) to the resulting text interpretation to identify the relevant information and store it in a database in the suitable format, or to fill forms as in MUC case. In this example, the IE is realized by transducers designed by Intex software, that insert XML labels in the text fragments when the syntactic and semantic conditions are verified. For example, the transducer in Figure 5 says states conditions, among others, there must be a noun phrase, subject of the verb and representing a protein (denoted by variable $1), and a noun phrase, direct object of the interaction verb (denoted by variable $2), representing a gene expression (denoted by variable $3). The gray boxes represent subtransducers. If all conditions are true, then XML protein, interaction and gene expression tags should be inserted (for example, see <protein>, <interaction> and <gene_expression> tags in the figure).

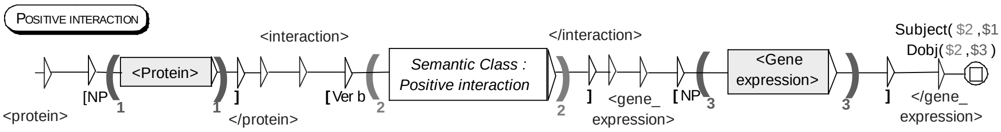


Figure 5. Example of extraction rule in the form of transducers for extracting gene interactions in functional genomics.

ML methods can help develop the knowledge bases needed for each step. For sentence filtering, discriminant keywords are learnable by classification methods such as naive Bayes or Support Vector Machines, [Marcotte *et al*., 2001], [Nedellec *et al*., 2001]. For building terminologies and ontologies from parsed corpora or assisting their design, unsupervised methods such as conceptual clustering are appropriate [Nedellec & Faure, 98]. The many methods designed for semantic class learning, query expansion, word sense disambiguation, or for building restrictions of selection are easily applicable to ontology and subcategorization frame learning. Then, predicate argument structures are learnable from subcategorization frame clustering or from semantically labeled corpora. Finally, extraction rules or automata (see Figure 5) are learnable from annotated corpora (see Figure 6) at the suitable level of linguistic interpretation (see Figure 4), [Sasaki & Matsuo, 2000]. The feasibility of such learning tasks from parsed corpora has been shown many times in the framework of specific domains such as scientific ones.

```
<SENTENCE                  name       = "2" >
          <INTERACTION
              id                 = "1"
              type   = "Y"
              Previous studies showed that <Agent1 type="Protein" func="Factor"> spoIIID </Agent1> <Interaction> is
              needed to produce </Interaction> <Target1 type="SigmaFactor">sigma K</Target1> [...]
          </INTERACTION>
```

Figure 5. Example of annotated sentence for IE rule learning. The highlights indicate the graphic attributes of the XML tags. For example, the regions tagged as "Interaction" are underlined, and the regions tagged as Agent are in bold.

Superficial approaches will not sufficiently resolve the problem of building IE systems for genomics. However, given the specificity of the language used in genomics texts, we can solve the task by combining ML from a corpus of annotated and un-annotated texts with syntactic–semantic analysis. Genomics provides demanding problems that will stimulate the development of more sophisticated approaches in IE. Although these aspects of extraction are not yet in the mainstream of IE research, this seems a promising direction not only for genomics but more generally for biology and for other perhaps less technical domains. Preliminary work in this area has produced encouraging results that we should now extend and deepen.

Acknowledgments

The author thanks her collaborators in the Caderige project, in particular Philippe Bessières, Gilles Bisson, Adeline Nazarenko and Mohamed Ould Abdel Vetah for their contribution to the scientific program presented in the paper. Caderige project is partially funded by the French inter-EPST program in bioinformatics.
This paper is a completed and revised version of the paper published by the IEEE Intelligent System Journal (May-June 2002).

References

Blaschke C., Andrade M. A., Ouzounis C. and Valencia A., “Automatic Extraction of Biological Information from Scientific Text: Protein-Protein Interactions,” *Proceedings of the International Symposium on. Molecular Biology* (ISMB’99), AAAI Press, USA pp. 60-67, 1999.

Craven M. and Kumlien J., “Constructing Biological Knowledge Bases by Extracting Information from Text Sources,” *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology* (ISMB-99), pp. 77-86, AAAI Press, USA, Heidelberg, Germany, 1999.

Faure D. and Nédellec C.,"Knowledge Acquisition of Predicate-Argument Structures from technichal Texts using Machine Learning" in Proceedings of *Current Developments in Knowledge Acquisition: EKAW-99*, p. 329-334, Fensel D. & Studer R. (Ed.), Springer Verlag, Karlsruhe, Germany, April 1999.

Freitag D., “Multistrategy Learning for Information Extraction,” *Proceedings of the 15th International Machine*

Learning Conference (ML'98), A. Danyluk (ed.) Morgan Kaufmann, pp. 100-107, Madison, Wisconsin, 1998.

Fukuda K., Tsunoda T., Tamura A. and Takagi T., “Toward Information Extraction: Identifying Protein Names from Biological Papers,” *Proceedings of the 3d Pacific Symposium on Biocomputing* (PSB’1998), 3:705-716 (http://www-smi.stanford.edu/projects/helix/psb98/), 1998.

Marcotte E. M., Xenarios I., and Eisenberg D., “Mining Literature for Protein-Protein Interactions,” *Bioinformatics Journal*, vol. 17, no. 4, pp. 359–363, Oxford University Press Applications, 2001.

Nédellec C., Ould Abdel Vetah M., and Bessières P., “Sentence Filtering for Information Extraction in Genomics: A Classification Problem,” *Proceedings of the International Conference on Practical Knowledge Discovery in Databases* (PKDD’2001), pp. 326–338, Springer Verlag, LNAI 2167, Freiburg, September, 2001.

Ono T., Hishigaki H., Tanigami A. and Takagi T., “Automated Extraction of Information on Protein-Protein Interactions from the Biological Literature,” *Bioinformatics Journal*, vol. 17, no. 2, pp. 155–161, Oxford University Press Applications, 2001.

Riloff E., “Automatically Constructing a Dictionary for Information Extraction Tasks,” *Proceedings of the 11th National Conference on Artificial Intelligence* (AAAI-93), pp. 811–816, AAAI Press/The MIT Press, Cambridge, Mass., 1993.

Rindflesh T., Tanabe L., Weinstein J.N. and Hunter L., “EDGAR: Extraction of Drugs, Genes and Relations from the Biomedical Literature,” *Proceedings of the 5th Pacific Symposium on. Biocomputing* (PSB’2000), (http://www-smi.stanford.edu/projects/helix/psb00/), pp. 514–525, 2000.

Sasaki Y. and Matsuo Y., “Learning Semantic-Level Information Extraction Rules by Type-Oriented ILP,” *Proceedings of the 18th International Conference on Computational Linguistics* (COLING-2000), Morgan Kaufmann, Saarbrüken, Germany, 2000

Sekimizu T., Park H. S. and Tsujii J., “Identifying the Interaction Between Genes and Gene Products Based on Frequently Seen Verbs in MedLine Abstracts,” *Genome Informatics*, pp. 62–71, Universal Academy Press, Tokyo, Japan, 1998.

Soderland S., “Learning Information Extraction Rules for Semi-Structured and Free Text,” *Machine Learning Journal*, vol. 34, pp. 233-272, 1999.

Thomas, J., Milward, D., Ouzounis C., Pulman S. and Caroll M., “Automatic Extraction of Protein Interactions from Scientific Abstracts,” *Proceedings of the 5th Pacific Symposium on Biocomputing* (PSB’2000), vol. 5, pp. 502–513, http://wwwsmi.stanford.edu/projects/helix/psb00/, 2000.

Relational Data Mining and Subgroup Discovery

Nada Lavrač

J. Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
nada.lavrac@ijs.si

Abstract. In Inductive Logic Programming (ILP), the recent shift of attention from program synthesis to knowledge discovery resulted in advanced relational data mining techniques that are practically applicable for discovering knowledge in relational databases. This paper gives a brief introduction to ILP, presents the state-of-the-art ILP techniques for relational knowledge discovery and outlines recent approaches to relational subgroup discovery.

1 Introduction

Inductive logic programming (ILP) [33, 36, 26, 12] is a research area that has its roots in inductive machine learning and logic programming. ILP research aims at a formal framework as well as practical algorithms for inductive learning of relational descriptions that typically have the form of logic programs. From logic programming, ILP has inherited its sound theoretical basis, and from machine learning, an experimental approach and orientation towards practical applications. ILP research has been strongly influenced also by Computational learning theory, and recently, also by Knowledge Discovery in Databases (KDD) [15] which led to the development of new techniques for relational data mining.

In general, an ILP learner is given an initial theory B (background knowledge) and some evidence E (examples), and its aim is to induce a theory H (hypothesis) that together with B *explains* some properties of E. In most cases the hypothesis H has to satisfy certain restrictions, which we shall refer to as a *bias*. Bias includes prior expectations and assumptions, and can therefore be considered as the logically unjustified part of the background knowledge. Bias is needed to reduce the number of candidate hypotheses. It consists of the language bias L, determining the hypothesis space, and the search bias which restricts the search of the space of possible hypotheses.

This paper first gives a brief introduction to ILP and presents a selection of recently developed ILP techniques for relational data mining [12], followed by an outline of recent approaches to relational subgroup discovery. The overview is restricted to techniques satisfying the strong criterion formulated for machine learning by Michie [31] that requires explicit symbolic form of induced descriptions.

2 State-of-the-art in ILP

This section briefly introduces two basic theoretical settings, gives pointers to successful ILP applications and presents recent technological developments in the area, categorized into the two main theoretical settings.

2.1 ILP problem specification

An inductive logic programming task can be formally defined as follows:

Given:

- a set of examples E
- a background theory B
- a language bias L that defines the clauses allowed in hypotheses
- a notion of *explanation*

Find: a hypothesis $H \subset L$ which explains the examples E with respect to the theory B.

This definition needs to be instantiated for different types of ILP tasks [36]. The instantiation will concern the representation of training examples, the choice of a hypothesis language and an appropriate notion of explanation. By explanation we here refer to an acceptance criterion of hypotheses: the hypothesis explains the data if it satisfies a certain user-defined criterion w.r.t. the data. We will discuss some formal acceptance criteria used in different ILP settings, but we also need to bear in mind that ILP aims at the induction of hypotheses that are expressed in an explicit symbolic form, that can be easily interpreted by the user/expert and may contribute to the better understanding of the problem addressed, ideally forming a piece of new knowledge discovered from the data.

2.2 ILP settings

The state-of-the-art ILP settings are overviewed below. For the underlying theory see [36, 37]. For a practical introduction to ILP see [26].

Predictive ILP *Predictive ILP* is the most common ILP setting, ofter referred to as *normal ILP*, *explanatory induction*, *discriminatory induction*, or *strong ILP*. Predictive ILP is aimed at learning of classification and prediction rules. This ILP setting typically restricts E to ground facts, and H and B to sets of definite clauses. The strict notion of explanation in this setting usually denotes coverage and requires global completeness and consistency.

Global completeness and consistency implicitly assume the notion of *intensional coverage* defined as follows. Given background theory B, hypothesis H and example set E, an example $e \in E$ is (intensionally) covered by H if $B \cup H \models e$. Hypothesis H is (globally) complete if $\forall e \in E^+ : B \cup H \models e$. Hypothesis H is (globally) consistent if $\forall e \in E^- : B \cup H \not\models e$.

Given the restriction to definite theories $T = H \cup B$, for which there exists a unique least Herbrand model $M(T)$, and to ground atoms as examples, this is equivalent to requiring that all examples in E^+ are true in $M(B \cup H)$ [36].

By relaxing the notion of explanation to allow incomplete and inconsistent theories that satisfy some other acceptance criteria (predictive accuracy, significance, compression), the predictive ILP setting can be extended to include learning of classification and prediction rules from imperfect data, as well as *learning of logical decision trees* [1]. In a broader sense, predictive ILP incorporates also *first-order regression* [22] and *constraint inductive logic programming* [40] for which again different acceptance criteria apply.

Descriptive ILP *Descriptive ILP* is sometimes referred to as *confirmatory induction*, *non-monotonic ILP*, *description learning*, or *weak ILP*. Descriptive ILP is usually aimed at learning of clausal theories [7]. This ILP setting typically restricts B to a set of definite clauses, H to a set of (general) clauses, and E to positive examples. The strict notion of explanation used in this setting requires that all clauses c in H are true in some preferred model of $T = B \cup E$, where the preferred model of T may be, for instance, the least Herbrand model $M(T)$. (One may also require the completeness and minimality of H, where completeness means that a maximally general hypothesis H is found, and minimality means that the hypothesis does not contain redundant clauses.)

By relaxing the strict notion of explanation used in clausal discovery [7] to allow for theories that satisfy some other acceptance criteria (similarity, associativity, interestingness), descriptive ILP can be extended to incorporate *learning of association rules* [2], *first-order clustering* [6, 13, 23], *database restructuring* [16, 42] *subgroup discovery* [45], *learning qualitative models* [21] and *equation discovery* [10].

An illustrative example Consider a problem of learning family relations where the predictive knowledge discovery task is to define the target relation `daughter(X,Y)`, which states that person `X` is a daughter of person `Y`, in terms of relations defined in background knowledge B. Let the training set E consist of positive and negative examples for the target predicate `daughter/2`. A positive example $e \in E^+$ provides information known to be true and should be entailed by the induced hypothesis. A negative example $e \in E^-$ provides information that is known not to be true and should not be entailed.

E^+ = {`daughter(mary,ann)`, `daughter(eve,tom)`}
E^- = {`daughter(tom,ann)`, `daughter(eve,ann)`}
B = {`mother(ann,mary)`, `mother(ann,tom)`, `father(tom,eve)`,
`father(tom,ian)`, `female(ann)`, `female(mary)`, `female(eve)`,
`parent(X,Y)` ← `mother(X,Y)`, `parent(X,Y)` ← `father(X,Y)`,
`male(pat)`, `male(tom)`}

If the hypothesis language L contains all definite clauses using the predicate and functor symbols appearing in the examples and background knowledge, a predictive ILP system can induce the following clause from E^+, E^- and B:

```
daughter(X,Y) ← female(X), parent(Y,X).
```

Alternatively, a learner could have induced a set of clauses:

```
daughter(X,Y) ← female(X), mother(Y,X).
daughter(X,Y) ← female(X), father(Y,X).
```

In descriptive knowledge discovery, given E^+ and B only, an induced theory could contain the following clauses:

```
← daughter(X,Y), mother(X,Y).
female(X) ← daughter(X,Y).
mother(X,Y); father(X,Y) ← parent(X,Y).
```

One can see that in the predictive knowledge discovery setting classification rules are generated, whereas in the descriptive setting database regularities are derived.

Other ILP settings There has been a suggestion [8] of how to integrate the two main settings of predictive and descriptive ILP. In this integrated framework the learned theory is a combination of (predictive) rules and (descriptive) integrity constraints that restrict the consequences of these rules.

Other ILP settings have also been investigated, the most important being *relational instance-based learning* [14]. Excellent predictive results have been achieved by the relational instance-based learner RIBL [14] in numerous classification and prediction tasks. Recently, *first-order reinforcement learning* [11] and first-order Bayesian classifier [18] have also been studied. Since these ILP settings do not involve hypothesis formation in explicit symbolic form, the developed techniques do not qualify as techniques for relational knowledge discovery.

3 Relational Data Mining Techniques

This section reviews the state-of-the-art relational data mining techniques most of which have already shown their potential for use in real-life applications. The overview is limited to recent Relational Data Mining developments, aimed at the analysis of real-life databases [27, 12]. These developments have a marketing potential in the prosperous new areas of Data Mining and Knowledge Discovery in Databases. It is worthwhile noticing that none of the reviewed techniques belongs to programming assistants which have a much smaller marketing potential and a limited usefulness for solving real-life problems in comparison with ILP data mining tools and techniques.

3.1 Predictive RDM techniques

Learning of classification rules. This is the standard ILP setting that has been used in numerous successful predictive knowledge discovery applications. The well-known systems for classification rule induction include Foil [39][1], Golem [35] and Progol [34]. Foil is efficient and best understood due to its similarity to Clark and Niblett's CN2. On the other hand, Golem and Progol are champions concerning successful ILP applications, despite the fact that they are substantially less efficient. Foil is a top-down learner, Golem is a bottom-up learner, and Progol uses a combined search strategy. All are mainly concerned with single predicate learning from positive and negative examples and background knowledge; in addition, Progol can also be used to learn from positive examples only. They use different acceptance criteria: compression, coverage/accuracy and minimal description length, respectively. The system LINUS [25, 26], developed from a learning component of QuMAS [32], introduced the propositionalization paradigm by transforming an ILP problem into a propositional learning task.

Induction of logical decision trees. The system Tilde [1] belongs to Top-down induction of decision tree algorithms. It can be viewed as a first-order upgrade of Quinlan's C4.5, employing logical queries in tree nodes which involves appropriate handling of variables. The main advantage of Tilde is its efficiency and capability of dealing with large numbers of training examples, which are the well-known properties of Tilde's propositional ancestors. Hence Tilde currently represents one of the most appropriate systems for predictive knowledge discovery. Besides the language bias, Tilde allows for lookahead and prepruning (according to the minimal number of examples covered) defined by parameter setting.

First-order regression. The relational regression task can be defined as follows: Given training examples as positive ground facts for the target predicate $r(Y, X_1, ..., X_n)$, where the variable Y has real values, and background knowledge predicate definitions, find a definition for $r(Y, X_1, ..., X_n)$, such that each clause has a literal binding Y (assuming that $X_1, ..., X_n$ are bound). Typical background knowledge predicates include less-or-equal tests, addition, subtraction and multiplication. An approach to relational regression is implemented in the system FORS (First Order Regression System) [22] which performs top-down search of a refinement graph. In each clause, FORS can predict a value for the target variable Y as the output value of a background knowledge literal, as a constant, or as a linear combination of variables appearing in the clause (using linear regression).

Inductive Constraint Logic Programming. It is well known that Constraint Logic Programming (CLP) can successfully deal with numerical constraints. The idea of Inductive Constraint Logic Programming (ICLP) [40] is to benefit from the number-handling capabilities of CLP, and to use the constraint solver of CLP to do part of the search involved in inductive learning. To this end a maximally discriminant generalization problem in ILP is transformed to

[1] A successor of Foil, the system Ffoil, can successfully be used for inducing relational definitions of functions.

an equivalent constraint satisfaction problem (CSP). The solutions of the original ILP problem can be constructed from the solutions of CSP, which can be obtained by running a constraint solver on CSP.

3.2 Descriptive RDM techniques

Learning of clausal theories and association rules. In discovering full clausal theories, as done in the system Claudien [7], each example is a Herbrand model, and the system searches for the most general clauses that are true in all the models. Clauses are discovered independently from each other, which is a substantial advantage for data mining, as compared to the learning of classification rules (particularly learning of mutually dependent predicates in multiple predicate learning). In Claudien, search of clauses is limited by the language bias. Its acceptance criterion can be modified by setting two parameters: the requested minimal accuracy and minimal number of examples covered. In another clausal discovery system, Primus [17], the best-first search for clauses is guided by heuristics measuring the "confirmation" of clauses. The Claudien system was further extended to Warmr [2] that enables learning of association rules from multiple relations.

First-order clustering. Top-down induction of decision trees can be viewed as a clustering method since nodes in the tree correspond to sets of examples with similar properties, thus forming concept hierarchies. This view was adopted in C0.5 [6], an upgrade of the Tilde logical decision tree learner. A relational distance-based clustering is presented also in [23]. An early approach combining learning and conceptual clustering techniques was implemented in the system Cola [13]. Given a small (sparse) set of classified training instances and a set of unclassified instances, Cola uses Bisson's conceptual clustering algorithm KBG on the entire set of instances, climbs the hierarchy tree and uses the classified instances to identify (single or disjunctive) class descriptions.

Database restructuring. The system Fender [42] searches for common parts of rules describing a concept, thus forming subconcept definitions to be used in the refurmulation of original rules. The result is a knowledge base with new intermediate concepts and deeper inferential structure than the initial "flat" rulebase. The system Index [16] is concerned with the problem of determining which attribute dependencies (functional or multivalued) hold in the given relational database. The induced attribute dependencies can be used to obtain a more structured database. Both approaches can be viewed as doing predicate invention, where (user selected) invented predicates are used for theory restructuring.

Subgroup discovery. The subgroup discovery task is defined as follows: given a population of individuals and a property of those individuals we are interested in, find the subgroups of the population that are statistically "most interesting", i.e., are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest. The system Midos [45] guides the top-down search of potentially interesting subgroups using numerous user-defined parameters.

Learning qualitative models of dynamic systems. The automated construction of models of dynamic system may be aimed at qualitative model discovery. A recent qualitative model discovery system [21], using a Qsim-like representation, is based on Coiera's Genmodel to which signal processing capabilities have been added.

Equation discovery. The system LAGRANGE [10] discovers a set of differential equations from an example behavior of a dynamic system. Example behaviors are specified by lists of measurements of a set of system variables, and background knowledge predicates enable the introduction of new variables as time derivatives, sines or cosines of system variables. New variables can be further introduced by multiplication.

Inductive databases. A tighter connection with deductive database technology has been recently advocated by Luc De Raedt [4, 5] introducing an inductive database mining query language that integrates concepts from ILP, CLP, deductive databases and meta-programming into a flexible environment for relational knowledge discovery in databases. Since the primitives of the language can easily be combined with Prolog, complex systems and behaviour can be specified declaratively. This type of integration of concepts from different areas of computational logic can prove extremely beneficial for RDMn the future. It can lead to a novel ILP paradigm of inductive logic programming query languages whose usefulness may be proved to be similar to those of constraint logic programming.

3.3 Some challenges in RDM research

ILP has already developed numerous useful techniques for relational knowledge discovery. A recent research trend in ILP is to develop algorithms implementing all the most popular machine learning techniques in the first-order framework. Already developed techniques upgrading propositional learning algorithms include first-order decision tree learning [1], first-order clustering [6, 23], relational genetic algorithms [20], first-order instance-based learning [14], first-order reinforcement learning [11] and first-order Bayesian classifier [18]. It is expected that the adaptation of propositional machine learning algorithms to the first-order framework will continue also in the areas for which first-order implementations still do not exist. This should provide a full scale methodology for relational data mining based on future ILP implementations of first-order Bayesian networks, first-order neural networks, possibly first-order fuzzy systems and other ILP upgrades of propositional machine learning techniques.

4 Relational Subgroup Discovery and Related Work

4.1 Relational Subgroup Discovery

In the newly emerging field of subgroup discovery two most important systems for discovering subgroups are Explora [24] and Midos [45]. The first system treats the learning task as a single relation problem, i.e., all the data are assumed to be available in one table (relation), while the second one extends this task to multi-relational databases, which is related to a number of other learning tasks [7, 30, 46], mostly in the ILP [12, 26]. In both systems the propositional (attribute-value) language is used to describe the induced hypotheses, i.e., discovered subgroups are defined as conjunctions of features (attributes values). The most important features of Explora and Midos concern the use of heuristics for subgroup discovery; the heuristics are outlined below.

We have developed a relational subgroup discovery system RSD [28] on principles that employ the following main ingredients: exhaustive first-order feature construction, elimination of irrelevant features, an implementation of a relational rule learner, the weighted covering algorithm and incorporation of example weights into the weighted relative accuracy heuristic, probabilistic classification, and area under ROC rule set evaluation.

As the input, RSD expects (a) a relational database containing one main table (relation) where each row corresponds to a unique *individual* and one attribute of the main table is specified as the *class* attribute, and (b) a mode-language definition used to construct first-order features.

The main output of RSD is a set of subgroups whose class-distributions differ substantially from those of the complete data-set. The subgroups are identified by conjunctions of symbols of pre-generated first-order features. As a by-product, RSD also provides a file containing the mentioned set of features and offers to export a single relation (as a text file) with rows corresponding to individuals and fields containing the truth values of respective features for the given individual. This table is thus a propositionalised representation of the input data and can be used as an input to various attribute-value learners.

An important feature of the RSD algorithm is the use of the weighted covering algorithm. In the classical covering algorithm of rule-set induction, only the first few induced rules may be of interest as subgroup descriptors with sufficient coverage, since subsequently induced rules are induced from biased example subsets, i.e., subsets including only positive examples not covered by previously induced rules. This bias constrains the population for subgroup discovery in a way that is unnatural for the subgroup discovery process which is, in general, aimed at discovering interesting properties of subgroups of the entire population. In contrast, the subsequent rules induced by the weighted covering algorithm allow for discovering interesting subgroup properties of the entire population.

The weighted covering algorithm is implemented in such a way that covered positive examples are not deleted from the current training set. Instead, in each run of the covering loop, the algorithm stores with each example a count how many times (with how many rules induced so far) the example has been covered. Weights of covered examples decrease according to the formula $e(i) = \frac{1}{i+1}$, where $e(i)$ is the weight of an example being covered i times.

A variant of the weighted covering algorithm has been used also in the context of the SD subgroup discovery algorithm [19], and in the CN2-SD subgroup discovery algorithm [28].

4.2 Measures of Interestingness

Various rule evaluation measures and heuristics have been studied for subgroup discovery, aimed at balancing the size of a group (referred to as factor g in [24]) with its distributional unusualness (referred to as factor p). The properties of functions that combine these two factors have been extensively studied (the "p-g-space"). Similarly, the weighted relative accuracy heuristic, defined as $WRAcc(Class \leftarrow Cond) = p(Cond).p(Class|Cond) - p(Class))$ and used in [44], trades off generality of the rule ($p(Cond)$, i.e., rule coverage) and relative accuracy ($p(Class|Cond) - p(Class)$). Besides such 'objective' measures of interestingness, some 'subjective' measure of interestingness of a discovered pattern can be taken into the account, such as actionability ('a pattern is interesting if the user can do something with it to his or her advantage') and unexpectedness ("a pattern is interesting to the user if it is surprising to the user") [41].

4.3 Subgroup Evaluation Measures

Evaluation of induced subgroups in the ROC space [38] shows classifier performance in terms of false alarm or *false positive rate* $FPr = \frac{FP}{TN+FP}$ (plotted on the X-axis) that needs to be minimized, and sensitivity or *true positive rate* $TPr = \frac{TP}{TP+FN}$ (plotted on the Y-axis) that needs to be maximized. The ROC space is appropriate for measuring the success of subgroup discovery, since subgroups whose TPr/FPr tradeoff is close to the diagonal can be discarded as insignificant. The standard approach is to use the area under the ROC convex hull defined by subgroups with the best TPr/FPr tradeoff as a quality measure for comparing the success of different learners.

Acknowledgments

This work was supported by the Slovenian Ministry of Education, Science and Sport, and the EU funded project Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise (IST-1999-11495). I am grateful to Luc De Raedt, Sašo Džeroski, Peter Flach and Dragan Gamberger for joint research in ILP and subgroup discovery.

References

1. H. Blockeel and L. De Raedt. Top-down induction of logical decision trees. Submitted to *DAMI, Special Issue on Inductive Logic Programming*, 1998.
2. L. Dehaspe, L. De Raedt. Mining association rules in multiple relations. *Proc. Seventh Int. Workshop on Inductive Logic Programming*, Springer, LNAI 1297, pp. 125–132, 1997.
3. L. De Raedt. A perspective on inductive logic programming. Invited lecture at *The Workshop on Current and Future Trends in Logic Programming*, Shakertown, to appear in Springer LNCS, 1999.
   Available at: `www.cs.kuleuven.ac.be/~lucdr/shaking.ps`.

4. L. De Raedt. A relational database mining query language. In *Proc. Fourth Workshop on Artificial Intelligence and Symbolic Computation*, Springer LNAI, 1998 (in press).
5. L. De Raedt. An inductive logic programming query language for database mining (extended abstract). In *Proc. JICSLP'98 post-conference workshop Compulog Net Area Meeting on Computational Logic and Machine Learning*, pp. 27–34, 1998.
6. L. De Raedt, H. Blockeel. Using logical decision trees for clustering. *Proc. Seventh Int. Workshop on Inductive Logic Programming*, Springer, LNAI 1297, pp. 133–140, 1997.
7. L. De Raedt, L. Dehaspe. Clausal discovery. *Machine Learning*, 26(2/3): 99–146, 1997.
8. Y. Dimopoulos, S. Džeroski and A.C. Kakas. Integrating Explanatory and Descriptive Induction in ILP. In *Proc. of the 15th International Joint Conference on Artificial Intelligence (IJCAI97)*, pp. 900–907, 1997.
9. S. Džeroski, L. De Haspe, B. Ruck, W. Walley. Classification of river water quality data using machine learning. In *Proc. of the Fifth Int. Conference on the Development and Application of Computer Technologies to Environmental Studies, Vol. I: Pollution Modelling.*, pp. 129–137. Computational Mechanics Publications, Southampton, 1994.
10. S. Džeroski, L. Todorovski. Discovering dynamics: From inductive logic programming to machine discovery. *Proc. Tenth Int. Conference on Machine Learning*, pp. 97–103, Morgan Kaufmann, 1993.
11. S. Džeroski, L. De Raedt, H. Blockeel. Relational reinforcement learning. In D. Page (ed.) *Proc. Eighth Int. Conference on Inductive Logic Programming*, pp. 11–22, Springer, LNAI 1446, 1998.
12. S. Džeroski, N. Lavrač (eds.) *Relational Data Mining.* Springer, 2001.
13. W. Emde. Learning of characteristic concept descriptions from small sets to classified examples. *Proc. Seventh European Conference on Machine Learning*, LNAI 784, pp. 103–121, Springer, 1994.
14. W. Emde, D. Wettschereck. Relational instance-based learning. *Proc. Thirteenth Int. Conference on Machine Learning*, pp. 122–130, Morgan Kaufmann, 1996.
15. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, eds. *Advances in Knowledge Discovery and Data Mining.* The MIT Press, 1995.
16. P.A. Flach. Predicate invention in inductive data engineering. *Proc. Sixth European Conference on Machine Learning*, Springer, LNAI 667, pp. 83-94, 1993.
17. P. Flach, N. Lachiche. Cooking up integrity constraints with Primus. Technical report, University of Bristol, 1998.
18. P.A. Flach and N. Lachiche. 1BC: A first-order Bayesian classifier. In *Proc. of the 9th International Workshop on Inductive Logic Programming (ILP'99)*, pp. 92–103, Springer LNAI 1634, 1999.
19. Gamberger, D. & Lavrač, N. (2002) Descriptive induction through subgroup discovery: a case study in a medical domain. In *Proc. of 19th International Conference on Machine Learning (ICML2002)*, Morgan Kaufmann, in press.
20. A. Giordana, C. Sale. Learning structured concepts using genetic algorithms. In *Proc. Ninth Int. Workshop on Machine Learning*, pp. 169–178, 1992.
21. D.T. Hau and E.W. Coiera. Learning qualitative models of dynamic systems. *Machine Learning*, 26(2/3): 177–212, 1997.
22. A. Karalič, I. Bratko. First order regression. *Machine Learning*, 26(2/3): 147–176, 1997.

23. M. Kirsten, S. Wrobel. Relational distance-based clustering. In D. Page (ed.) *Proc. Eighth Int. Conference on Inductive Logic Programming*, pp. 261–270, Springer, LNAI 1446, 1998.
24. Klösgen, W. (1996) Explora: A multipattern and multistrategy discovery assistant. In U.M. Fayyad, G. Piatetski-Shapiro, P. Smyth and R. Uthurusamy (Eds.) *Advances in Knowledge Discovery and Data Mining*, 249–271. MIT Press.
25. N. Lavrač, S. Džeroski and M. Grobelnik. Learning nonrecursive definitions of relations with LINUS. In *Proc. Fifth European Working Session on Learning*, pp. 265–281. Springer.
26. N. Lavrač and S. Džeroski. *Inductive Logic Programming: Techniques and Applications.* Ellis Horwood, 1994.
27. N. Lavrač, S. Džeroski and M. Numao. Inductive logic programming for relational knowledge discovery. *New Generation Computing* 17 (1): 3–23, 1999.
28. N. Lavrač, F. Železný, P. Flach. Relational subgroup discovery: A propositionalization approach through first-order feature construction. *Proc. Int. Conference on Inductive Logic Programming, ILP-2002.* Springer, 2002 (in press).
29. N. Lavrač, P. Flach, B. Kavšek, L. Todorovski. Rule induction for subgroup discovery with CN2-SD. *Proc. Integrating Aspects of Data Mining, Decision Support and Meta-Learning*, Workshop at the ECML/PKDD-2002 Conference, in press, 2002.
30. H. Mannila, H. Toivonen. On an algorithm for finding all interesting sentences. In R. Trappl, ed., *Proc. Cybernetics and Systems'96*, pp. 973–978, 1996.
31. D. Michie. Machine learning in the next five years. *Proc. Third European Working Session on Learning*, pp. 107–122. Pitman, 1988.
32. I. Mozetič. Learning of qualitative models. In I. Bratko and N. Lavrač (eds.) *Progress in Machine Learning*, pp. 201–217. Sigma Press, 1987.
33. S. Muggleton, ed. *Inductive Logic Programming.* Academic Press, 1992.
34. S. Muggleton. Inverse Entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4), 1995.
35. S. Muggleton, C. Feng. Efficient induction of logic programs. *Proc. First Conference on Algorithmic Learning Theory*, pp. 368–381. Ohmsha, Tokyo, 1990.
36. S. Muggleton, L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming* 19/20: 629–679, 1994.
37. S.H. Nienhuys-Cheng, R. de Wolf. *Foundations of inductive logic programming.* Springer LNAI 1228, 1997.
38. Provost, F. & Fawcett, T. (2001) Robust classification for imprecise environments. *Machine Learning*, 42(3), 203–231.
39. J.R. Quinlan. Learning logical definitions from Relations. *Machine Learning* 5:239-266, 1990.
40. M. Sebag, C. Rouveirol. Constraint Inductive Logic Programming. In L. De Raedt, ed., *Advances in Inductive Logic Programming*, pp. 277–294, IOS Press, 1996.
41. A. Silberschatz, A. Tuzhilin. On Subjective Measure of Interestingness in Knowledge Discovery. In *Proc. First International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 275–281, 1995.
42. E. Sommer. Rulebase stratifications: An approach to theory restructuring. *Proc. Fourth Int. Workshop on Inductive Logic Programming*, GMD-Studien 237, pp. 377-390, 1994.
43. A. Srinivasan, R.D. King, S. Muggleton, M.J.E. Sternberg. The Predictive Toxicology Evaluation Challenge. In *Proc. Fifteenth Int. Joint Conf. on Artificial Intelligence*, pp. 4–9, Morgan Kaufmann, 1997.

44. L. Todorovski, P. Flach, N. Lavrač. Predictive Performance of Weighted Relative Accuracy. In Zighed, D.A., Komorowski, J. and Zytkow, J., editors, *Proc. of the 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2000)*, pp. 255–264, Springer, 2000.
45. S. Wrobel. An algorithm for multi-relational discovery of subgroups. *Proc. First European Symposium on Principles of Data Mining and Knowledge Discovery*, pp. 78–87. Springer, 1997.
46. S. Wrobel, S. Džeroski. The ILP description learning problem: Towards a general model-level definition of data mining in ILP. In K. Morik and J. Herrmann, eds, *Proc. Fachgruppentreffen Maschinelles Lernen (FGML-95)*, 44221 Dortmund, Univ. Dortmund, 1995.

Ontology Enrichment with Texts from the WWW

Andreas Faatz, Ralf Steinmetz

KOM - Multimedia Communications Lab
Darmstadt University of Technology, Merckstr. 25, 64283 Darmstadt, Germany
{afaatz, rst}@kom.tu-darmstadt.de

Abstract. The following paper explains, how we can enrich an existing ontology by mining the WWW. The use of such an ontology may be manifold, for example as a component of information systems or multimedial repositories. The enrichment process is based on the comparison between statistical information of word usage in a large text collection, a so called text corpus, and the structure of the ontology itself. The text corpus will be constructed by using the vocabulary from the ontology and querying the WWW via Google.
We define similarity measures by optimising their parametrisation and examine the central properties of the enrichment approach - along with the presentation and evaluation of experimental results. Parametrisation of a similarity measure means assigning weights to each word collocation feature we first check in the text corpus and thereafter integrate into the representation of a word or a concept.

1 Introduction

Automatic thesaurus and ontology construction dates back from the last three decades [1]. Our approach is a further development of methods to construct the whole ontology automatically. In contrast to these approaches our algorithm can only be applied, if we enrich an existing ontology instead of fully constructing the ontology.
The following paper focuses on requirements for the semi-automatic enrichment of medical ontologies based on the statistical information of word usage. An ontology is a structured network of concepts from an knowledge domain and interconnects the concepts by semantic relations and inheritance. [2] gives a precise technical definition of an ontology, that we will refer to throughout this paper:

Definition 1: An *ontology* is a 4-tuple $\Omega := (C, \text{is\_a}, R, \sigma)$, where C is a set we call *concepts*, *is_a* is a partial order relation on C, R is a set of relation names and $\sigma : R \longrightarrow \wp(CXC)$ is a function [2].
Throughout this paper we assume that a concept has a character string as a descriptor. This character string may be a word or a phrase.

For our purposes we will neglect R as well as σ and focus on *is_a* as the particular relation, which is responsible for superconcept-subconcept dependencies. For example *bacteria* is a superconcept of the concept *pathogenic bacteria*. Whenever we talk of ’relations’ or ’relational paths’ in the following sections, we refer to the *is_a* relation. We also define

Definition 2 : We call the restriction of an ontology $\Omega := (C, \text{is\_a}, R, \sigma)$ to $(C, \text{is\_a})$ the *hierarchical backbone* of Ω.

Ontologies give a formal representation and conceptualisation of a knowledge domain, which is useful for the administration of large multimedial resource collections: if ontologies reflect an agreement of a group of experts and are rich enough in the sense of a sufficient number of concepts, ontologies are able to handle information exchange across the borders of one expert's vocabulary. For example, one could ask an ontology to return the names of all bacteria causing diarrhoea and in this way access domain knowledge without the barrier of finding out the names of the particular bacteria by reading texts from the domain of infectiology.

Naturally the construction of an ontology is hard and expensive, as one has to train domain experts in formal knowledge representation. This is the motivation behind the idea, that for a given ontology we focus on finding new concepts automatically. Those new concepts are propositions, which extend the given ontology. For this we

- use a special text corpus derived from WWW search results
- detect a set of candidate concepts from the corpus
- finally select a subset of those candidate concepts ranking their similarity to concepts already existing in the given ontology.

The final selection ends up in new concepts for the ontology to be proposed to a (human) ontology engineer.

The concepts have one or more descriptors, which are words or phrases from natural language. This implies that we develop our method finding suitable definitions for the semantic similarity of words or ordered sets of words.

The paper is organised as follows: in section 2.1 we describe our approach informally, whereas in 2.2 we give a survey on formalisation, which is explained in detail in section 2.3. Section 3 deals with the experimental results on two very different kinds of text corpora and especially on mining propositions from Google search hits. Section 4 discusses related work. At last section 5 summarises and points to future work.

2 Enrichment Approach

2.1 Overview

Similarity between words is a topic from the theory of word clustering algorithms and requires statistical information about the context, in which the words are used. Many approaches check collocation features of the words in large text corpora, such that a word is represented by a large vector. The vector has entries communicating, how often a collocation feature was fulfilled in the corpus. The vectors are sparse [3]. The notion of similarity definitions by vector representations normally does not assign a weight to every single dimension of the vectors. In this paper we argue, that this is possible by a soft method using the information already defined in the given ontology. The influence of the ontological structure on the word (-vector) similarities results in an optimisation problem, which determines which dimension in the word (-vector) representation is influential for the similarity computation. The following definition should clarify, what a collocator is.

Definition 3: Let a word *w* be given. A *collocator* of *w* is a word, which occurs together with *w* due to a predefined rule in a text collection (text corpus).

Thus for example in the phrase '*Medical ontology enrichment in the k-med project*', '*enrichment*' and '*medical*' would be collocators for predefined rules like for instance 'maximal distance 5 words' or also 'occurrence in the same sentence'.
The way we include the ontological information from $\Omega := (C, \text{is\_a}, R, \sigma)$ may be guided by different heuristics on a numerical interpretation of is_a, *R* and σ. For example the abstraction level of a concept, the interconnection by relations, relational paths and their lengths or the local granularity of the modelling can establish distance measures on a given ontology Ω.
Our goal is a comparison of this distance measures to the information about collocators in a text corpus.

2.2 Enrichment as an optimisation problem

The core idea of our approach is computing enrichment rules, which do not contradict the distance information already given by the ontology we want to enrich.
We first have to state a basic assumption.

Assumption 1: There exists a consistent distance measure d expressing semantic distances between the concepts in Ω. The distance measure is based on the relational interconnections between the concepts in the hierarchical backbone of Ω.

By consistency we mean, that *d* underlies some characteristic heuristics: a long relational path between two concepts rises the distance between them, the abstraction level in the hierarchical backbone influences the distance measure *d* as well as the number of concepts, which are subconcepts to the same superconcept. Both abstraction and the number of siblings rise the distance. The distance measure *d*, which we will from now on denote by *d(x,y)* for concepts *x* and *y* from Ω also differentiates between generalisation and specialisation in an ontology. We showed in [4] that such distance measures exist indeed. Thereby our notion of 'consistent' does not necessarily imply a good enrichment quality, but just means, that the above heuristics are fulfilled. The quality of the heuristics and the resulting distance measures have to be judged by the results of enrichment experiments.
We also assume a text corpus ζ to be given and determine the ordered set *K* of the *n* most frequent collocators which cooccur with at least two concepts from Ω. Let $v(x) \in R^n$ be a vector; the *i*-th entry of this vector *v(x)* expresses, how often the descriptor of the concept $x \in \Omega$ was a collocator in ζ with the *i*-th element of *K*.

Let now f_k be a component-wise monotonic function of the dissimilarity D between two concepts *x* and *y* from Ω . For the dissimilarity D we postulate, that it must be computable from the vectors *v(x)* with concepts *x* from Ω .
The parametrisation $k = (k_1, \ , \ , k_n)$ just weighs each collocation feature positively, it indicates, how strong the *j*-th dimension is involved in the dissimilarity computation. The optimisation process consequently fits the average of the f_k *(* D (*v(x), v(y)))* for possible $k = (k_1, \ , \ , k_n)$ with each $k_i \geq 0$, to the distances *d(x,y)* for each pair of concepts from Ω .
To sum it up briefly, the optimisation establishes a dissimilarity measure, which is as near as possible to the distance measure *d* in Ω . In the next section we present a formalisation of the algorithm. We decided to explain the details of this formalisation to make a repetition of experiments with the algorithm possible.

2.3 Formalisation of the algorithm

A *distance measure* on Ω is a function *d:* (*C* X *C*) $\longrightarrow$ [0,1]. Examples of distance measures are:

1) *d(x,y)=* e^s , where *e* is Euler's constant and *s* denotes the number of steps along the shortest relational path between the concepts *x* and *y*. This distance definition corresponds to the heuristics, that long relational paths rise the distance between given concepts.
2) *d(x,y)=1,* if there exists a relation between the concepts *x* and *y* and *d(x,y)=0,* if there does not exist a relation between the concepts *x* and *y*. This definition has only a reasonable application to ontologies, if the transitivity of the is_a-Relation and the concatenation of different relations from *R* is clearly stated in the a of axioms of Ω .
3) [5] defined criteria for similarity measures in thesauri, which in turn can be applied to distances in the hierarchical backbone of the ontology Ω := (*C*, is_a, *R* , σ).
[4] showed, that there is an infinite number of distance measures on the hierarchical backbone of an ontology fulfilling more restrictive characteristics than 1) and 2). For further details we have to refer to [4].

A *text corpus* ζ is a collection of text documents written in exactly one natural language. We assume ζ to be electronically available. From a text corpus we define a set of words or phrases to be the candidate concepts. A *proposition* for the ontological enrichment is a word or a phrase from ζ, which is used similarly to the concepts from the given ontology. Candidates are to be predefined, for example as all nouns occurring in ζ . Note that ζ might also be extended by additional text material. This may happen during or after the application of the enrichment algorithm.
A *rule set* ρ is a finite set of linguistic properties, each of which can be tested in terms of its fulfilment frequency in the text corpus. In our case we will always consider collocation properties for the rule set ρ .
The entries m_{ij} of a *representation matrix* *M*(*C*, ρ, ζ) list, how often the *j*-th property from ρ was fulfilled in ζ for the descriptor the *i*-th concept from *C*.
The enrichment algorithm processes information available from ζ and Ω . It computes an optimal solution for the problem of fitting the distance information among the concepts expressed by Ω and the dissimilarity information between words or phrases to be extracted from the word usage statistics considering ζ .

Let us assume a given $M(C, \rho, \zeta)$. We search for a set $k = \{k_1, \cdot, k_n\}$ of non-negative reals with $|k| = |\rho|$, which will be called *configuration* of the rule set ρ. Each k_i corresponds to a rule ρ_i.
The configuration k decides about the quantities of dissimilarity we derive from $M(C, \rho, \zeta)$.
The *Kullback-Leibler divergence* generally measures the dissimilarity between two probability mass functions [6] and was applied successfully to statistical language modelling and prediction problems [7]. The Kullback-Leibler *D(x,y)* divergence for two words *x, y* is defined as

$$D(x, y) = \sum_{w} P(w|x) \log \frac{P(w|x)}{P(w|y)} \tag{1}$$

In the basic version of the Kullback-Leibler divergence, which is expressed by formula (1), *w* is a linguistic property and $P(w|x)$ is the probability of this property being fulfilled for the word *x*. In the sum indicated by formula (1), *w* ranges over all linguistic properties one includes in a corpus analysis. In our case the frequencies of observing the collocation properties are denoted by $M(C, \rho, \zeta)$. We change (1) in such a way, that *k* weighs the influence of each property *w*
:

$$D_k(x, y) = \sum_{w} k(w) P(w|x) \log \frac{P(w|x)}{P(w|y)} \tag{2}$$

with $k(w) \in k$ in our case
Considering our representation matrix notation $M(C, \rho, \zeta)$ we obtain

$$P(w|x_i) = \sum_{l=1}^{|\rho|} \left[\frac{m_{il}}{\sum_{n=1}^{|\rho|} m_{in}} \right] \tag{3}$$

Let us clarify the notation of formula (3):
x_i denotes the *i*-th concept from *C*. Correspondingly in (3) the m_{il} are the matrix entries in $M(C, \rho, \zeta)$ in the row expressing the collocation properties of x_i. With this notation $k(x_i) = k_i$ holds. In that sense, we will be able to determine an optimal $k = \{k_1, \cdot, k_n\}$.
Taking the distances from the ontology Ω as an input, which should be approximated by the $D_k(x, y)$ as well as possible, the question of finding an optimal configuration *k* reduces to the question:
which configuration *k* minimises the average squared error expressed by the differences

$$(d(x, y) - D_k(x, y))^2 \quad ?$$

Finally we present a formulation of this question in terms of a quadratic optimisation formula. Searching for an optimal *k* means searching for a minimum of the following fitness expression:

$$\min_{k} \sum_{i = 1}^{|C|} \sum_{i = 1}^{|C|} (d(x_i, x_j) - D_k(x_i, x_j))^2 \tag{4}$$

where $k = \{k_1, \ , \ , k_n\}$ and $k_l \geq 0$ for all $k_l \in k$. Note that we minimise over the set of all configurations, that means over all possible *k*. We now explain, which words phrases are propositions for the ontological enrichment.

Once we optimised formula (4) we obtain the configuration in need to compute all the distance measures between all the concepts from Ω and the candidates. We apply an enrichment step starting with the optimal similarity measures $D_k(x, y)$.

We only take into concern the $D_k(x, y)$ with $x \in C$ and a candidate *y*. If such a distance between a formerly known concept (i.e. its descriptor) and a candidate (i.e. a word from the corpus) formerly unknown to Ω is lower than a predefined threshold, *y* proposition to enrich Ω. A suitable threshold can for example be defined from the average of the distances *d(x,y)* where *x ~ y* holds for some $\sim \ \in R$.

Additionally the $D_k(x, y)$ with $x \in C$ and a candidate *y* carry even more information, namely an optimal *placement* of the candidate concepts. The candidate concepts and the concepts from Ω can be presented together, if a candidate turns out to be a proposition. This simplifies the knowledge engineer's understanding of how the candidate concepts evolved and in which semantic area of Ω they might belong.

3 Experimental results

3.1 Basic input: ontology and corpus

The ontology Ω we enriched in our experiments is a modelling carried out by a medical expert during the first phase of the k-med project. K-med is an abbreviation for ’knowledge based multimedia medical education’. This project tries to collect multimedial medical learning resources and for the sake of reuse describe the educational resources by applying a metadata scheme and a common medical ontology [8].

The ontology contains the most abstract concept *disease OR symptom* along with the subconcepts *measles*, *German measles, diarrhoea, intestinal infection*. *Diarrhoea* itself has the subconcepts *aqueous diarrhoea* and *sanguinary diarrhoea*. The ontology may be viewed as an incomplete test ontology for several reasons: it only uses hierarchical relations, the superconcept and subconcept relations and also the knowledge domain are not fully clarified (such that a construction like *’disease OR symptom’* with a logical *OR* becomes necessary) and at least one additional abstraction level (a concept like ’*infections*’) could make the modelling clearer. In fact, this ontology is just a part of a larger ontology under construction. It is based on the subjects, which have to be tought during the first semesters of medical education in Germany.

To sum it up, we enriched this intuitively modelled ontology in the sense of [9], without deductive or inductive rules or axioms. From our point of view this enrichment of incomplete ontologies is, along with the extension of thesauri and catalogues, the main application area of our approach. Furthermore following [9], such a rather informal ontology represents a situation, where machine learning techniques should support the knowledge engineer.

The size of the ontology was kept small for the sake of a rapid application and evaluation. For the experiments presented in the remainder of this paper, it was easier to reduce the possible interdependencies by referring to this ontology chunk containing seven concepts.

As larger ontologies can be segmented to smaller ones - for instance to speed up the computation - we consider the enrichment of the ontological chunk a good starting point for experiments.

The computation of the represenation matrix and the derivation of the optimisation problem was carried out by an implementation of our own. For the sake of a possible later connection of this component to other existing ontology tools and the linguistic workbench TATOE [10] we used Smalltalk as the language to implement the algorithm. The quadratic optimisation (4) itself was carried out by the ampl-solver [11].

For our first experiments, we used a general, but very large (about 28.700.000 sentences) newspaper corpus available at [12]. The corpus ζ can be queried by on-line query tools, which also provide stemming and lemmatisation techniques. All queries are collocation queries determining, whether two words were used in ζ at a distance of predefined size. Although these experiments produced bad enrichment results from the medical expert's point of view, very important meta-properties of the algorithm, as compression of the rule set and a stability of the algorithm were found.

A second experiment was based on the search results of the web search engine Google [13]. We passed each descriptor of a concept to Google [13] and converted the documents belonging to the 10 search hits with the highest ranking into a text corpus. In contrast to the newspaper corpus this corpus is more specialised, consisting of 70 documents with 135.166 words and 15.570 sentences. Because of a restriction of the concordancer freeware in use (Wconcord, Darmstadt University of Technology) we did not apply stemming and lemmatisation to the specialised corpus we gained from the WWW. In our Smalltalk implementation we included a stop list consisting of auxiliary verbs, conjunctions, personal pronomina and prepositions.

For the rule set ρ we always tested, how often a collocation at maximum distance five tokens, but in the same sentence took place.

All of our enrichment results can be found in table 1. In the column at the very left the reader will find the concept from Ω, in the middle column the concepts from the general newspaper corpus proposed to the concept from Ω. At the right we find the propositions to Ω from the special corpus.

In both experiments a candidate became a proposition, if we computed a dissimilarity below 0.5. The reason for the choice of this threshold was, that a path of length 2 in the hierarchical backbone of the test ontology led to a distance average of 0.5. All experiments were carried out in German, so we show translations here.

Table 1: enrichment results

concept from Ω	general corpus (28.700.000 sentences)	special (www) corpus (15.500 sentences)
disease OR symptom	*loss* *illness* *infections body* *leg* *wound* *animals* *vaccination* *fever* *combat*	
intestinal infection		*medical doctor* *cause*
diarrhoea	*ailment* *epidemic* *cough* *fever* *vaccination* *infections*	*vomit* *stomach ache* *nausea* *fever* *medical doctor*
measles		
German measles		
sanguinary diarrhoea		*vomit* *stomach ache* *nausea* *fever* *can*
aqueous diarrhoea		*vomit* *stomach ache* *nausea* *fever* *can*

With ’*stomach ache*’ and ’*medical doctor*’ in German we did not propose word groups but composita (’*Bauchschmerzen*’ and ’*Arzt*’ in German).

3.2 Results for general corpora

We refer to the enrichment results depicted in table 1. As the rule set ρ we used collocation in the same sentence at maximal distance of five tokens in ζ.
Although we find some concepts like '*infection*' which is a missing abstraction level in the ontology, the enrichment results from general corpora are poor. We retrieve overly general propositions like 'body' and even flaws like the proposition of 'wound', 'animal' or 'leg'. These flaws occured especially with candidates, which only had one property from ρ.
Another problem occurs with special concepts - as expected, even a very large newspaper corpus does not contain enough information to get propositions for the subconcepts of diarrhoea.
Although we faced these problems, the experiments for general corpora were worthwhile, because we identified two considerable meta-properties of the approach: *stability* and *compression*.

a) *Stability*

Before dealing with the core of the enrichment - the proposal of concepts - we tested the inner stability of the approach and pruned $M(C, \rho, \zeta)$, which was a square matrix of size 102, in two different ways. If $M(C, \rho, \zeta)$ contained not enough information, then different pruning strategies would bare the danger of destroying the enrichment process, which means leading to inconsistent optimal configurations or enrichment results.

Table 2: stability

first strrategy	second strategy
suffer	*suffered*
diseases	*fever*
illness	*measles*
	hepatitis
die	*died*
pregnancy	*pregnancy*
percent	*percent*
stomach	

The first pruning strategy was keeping only the ten largest entries per row, resulting in a 102 X 34 matrix $M(C, \rho_{first}, \zeta)$. The second pruning strategy only kept the entries $c_{ij} \in M(C, \rho_{second}, \zeta)$ with c_{ij} >10, resulting in a 102 X 63 matrix. With both ma-

trices we set up the optimization procedure, solved expression (4) respectively and derived two optimal configurations k_{first} and $k_{\sec ond}$. The collocators belonging to rules with nonzero weights are listed in table 2. Both strategies mean collocation at maximal distance five words with a descriptor from the respective column of table 2.
Let us comment the collocators remaining from the two pruning strategies and the optimization. In table 2 we listed the collocators in such a way, that we immediatly see the relation between the two sets. Some of the collocators do not differ at all, some of them are only different in terms of the grammatical context they stem from (for example '*suffer*' and '*suffered*'), some of them obviously carry a semantic relation ('*diseases*' and '*illness*' on the one hand and their more specialised pendants '*feaver*', '*measles*' and '*hepatitis*' on the other hand). The only collocator without any direct relative in the other set is '*stomach*', so we state, that the analysis of the resulting collocator sets of nonzero weight does not show any inner contradiction in the approach, the representation matrix in our case seems to be stable and even carrying redundant information.

b) *Compression of* ρ

With both pruning strategies only a few properties ρ_i from ρ achived a corresponding weight k_i from *k*, with $k_i > 0$. This compression of the rule set also occured for different definitions of the ontological distance *d.* We assume, that this compression is closely related to the sparsity of our representation matrix and to the structure of the optimisation formula (4). The reason for this assumption are further experiments with artificially and randomly generated matrices, which we used as pseudo-representation matrices with sparsity structures similar to the ones of $M(C, \rho_{first}, \zeta)$ and $M(C, \rho_{\sec ond}, \zeta)$. As these experiments ended in a similar compression, we will search for a proper mathematical reason why this reduction of influential features with nonzero weight takes place.

3.3 Results for a special corpus based on Google hits

As we mentioned, we passed each descriptor of a concept to a web search engine and converted the first 10 hits of the Google search result into text files, removing the HTML-specific tagging. The results can be found in table 1.
As the rule set ρ we used again collocation at maximal distance of five words in ζ. For pruning reasons from our representation matrix we kept only properties from ρ, which were fulfilled for at least two concepts from the small test ontology. This resulted in a representation matrix with 292 columns. This means, that - for the special corpus - we initially found significantly more rules than with the common corpus, but after the solution of (4) we obtained 12 properties from ρ with a nonzero weight. These were distance five in the same sentence with *chronic, infection, because of, seldom, diarrhoea, vaccinate, pneunomia, virus*.
The choice of the candidates for the enrichment was driven by the observation from the previous experiments: propositons with only one nonzero property induced flaws to the enrichment. Consequently we accepted candidates, which at least fulfilled two of the 12 remaining properties from ρ.

The main results of the experiment with the special corpus crafted from web search hits are

- enrichment of the special concepts *sanguinary diarrhoea* and *aqueous diarrhoea*
- identification of a group of symptoms (*vomit, stomach ache, nausea, fever)* as propositions
- lack of propositions for *measles* and *German measles*

The flaws in this enrichment are *can* which should actually be a member of the stoplist and *medical doctor* which is at least too overgeneral.

3.4 Discussion of the results

As a second general observation we state, that the main flaws identified during the evaluation in this section come from candidates, which share only one feature with a concept from Ω. We conclude, that a possible way to handle this may be an additional tuning of the definition of *d*. A potential technique for this is generating artificial usage profiles, which do not at all reflect real words, and searching for a measure, which properly discriminates between real words and randomly constructed feature sets.

There exist subjective and objective ways of evaluating the results. Roughly speaking the objective evaluation methods base on reference ontologies, the subjective evaluations are based on expert interviews [14].

The question posed in the subjective evaluation was:

Consider the ontology Ω and the table of propositions from table 1 to be given. Which strategy performs better? Which aspects of the enrichment results are positive, which ones are negative?

The subjective evaluation clearly showed, that the enrichment with the special corpus performs better, as there are less flaws like *wounds* or *leg* and also less overgeneralisations like *illness*. In addition to this, the results of the specialised enrichment are easier to perceive, as there are not too many propositions and the good propositions were more precise.

The fact that our specialised enrichment with a web-based corpus performs better is not as trivial as it seems. For instance, the symptom group *vomit, stomach ache, nausea, fever* was also on the candidate list for the general corpus experiments - but the collocation information was insufficient, although the corpus was very large.

Comparing the experiments, possible causes for lacking propositions for *measles* and *German measles* may be found in the structure of out test ontology. It contains more information about diarrhetic diseases. At least the special text corpus must be balanced, a preprocessing step we will include in future experiments.

Other good candidates (like ’*virus*’) did not become propositions in the special corpus experiment, as we did not use stemming and lemmatisation. This introduces all inflections of verbs to the candidate set. Instead of this, the inflections should be unified to one candidate.

The main goal of a series of evaluations should be finding a correlation between subjective and objective measures. If such a correlation exists, even imprecise objective evaluation measures can give answers to the quality of parameter tuning or candidate strategies. The reason for this is, that the objective evaluation measures have to correlate ordinally but not cardinally with the subjective ones.

Objective evaluation measures we are developing are guided by the notion of precision in document retrieval [1]. Naturally we need a larger series of experiments to detect a possible correlation between objective and subjective evaluation measures.

4 Related work

Two main branches of automated ontology construction by natural language processing in general and checking collocators in our special case may be identified: those, which base the similarity of concepts or their descriptors on syntactic criteria or collocation directly (we will refer to those as first type) and those, which take statistic samples of the features of a concept or its descriptor. For example the first type declares concepts or their descriptors as similar, when they often occur together in one sentence. The works of [15]and [16] are examples of this type of enrichment.

The second type declares words as similar, if they are used in an similar context. For example, if a word *w* is used with a word *v* in the same sentence very often, and also a word *u* is used with a word *v* in the same sentence very often, then *w* and *u* would be similar according to the assumptions of the second type approaches, even if *w* and *u* never appear in the same sentence all over the text corpus. Note the significant difference between the approaches: the first type would state a high similarity between *w* and *v*, also between *u* and *v*. Representatives of the second type are the works of [17], [18], [19] and also [20]. The latter ones use syntactic parsing instead of pure collocation information. [20] moreover does not assume identity between concepts and their descriptors as we did in this paper.

If we try to group our approach among the first or second type approaches, we can clearly state that the way we define word similarities and dissimilarities is due to the second type as we pointed out in the previous section. Nevertheless the way we restrict our view to certain candidate concepts and their respective features also refers to the first type: a candidate concept is a concept, of which we determine similarities or dissimilarities to all ontological concepts. As for almost natural computational restrictions we are not able to compute the dissimilarities for all the words from a corpus, we must restrict our observations. Our candidate strategies in section 3 explain possible restrictions in detail. They are influenced by ideas of the first type.

Although - except [21] - none of the related works mentioned here directly focuses on ontology enrichment from the WWW, all these works have in common, that their automated construction features can be extended to our enrichment goals of proposition identification and placement.

[15] used pure collocation information for gaining new concepts, but also focuses on qualitative issues of the collocations to derive statements about relations and the behaviour of the relations.

[16] focused on the so-called salient words, which are able to disambiguate word meaning very well. In a way [16] is also a extending special form of ontology, namely a thesaurus. In contrast to our approach his focus is on disambiguation, which was further

developed by [21], but with web resources. In our approach a concept can be proposed to two different concepts, but may also disambiguate, depending on our ontological distance measures. Especially the identification of the symptom group *vomit, stomach ache, nausea, fever* propositions for several concepts from Ω would be simply impossible with the approaches of [16] or [21], as they found on discriminating descriptors. They achieve this by a test which detects the mostly diverging contexts of concepts of a given ontology (or thesaurus).

[19] designed the Mo'K workbench for word clustering and building hierarchies from the clusters in a second step. We also implemented a feature based environment, and our goal is even more specialised than word clustering. However we share the opinion with [19], that collecting many features and assigning weights to the features is an excellent basis for similarity definitions. Our implementation is in Smalltalk, whereas Mo'K uses C. Smalltalk remains a possible language, as we end up in very condensed representations with a few features of nonzero weight (compare section 3.2). Generally spoken, in contrast to our work (which systematically computes an optimal configuration *k* for a rule set ρ) [19] do not explicitly offer a strategy for choosing the weights.

An approach related to [19], but more founded on collocation networks and determining artificially specialised corpora can be found in [18] and a comparison of the performance of specialised and common corpora in word clustering can be found in [Asium].

Finally [17] experimented with word similarities expressed by an unsupervised neural network algorithm, the Kohonen map [23], but also for clustering, not for enrichment goals. Our evaluation methods result from the enrichment goal and could also be a basis for an evaluation of the Kohonen maps in word clustering, a task desired by [22].

5 Conclusion and future work

We presented a method for ontology enrichment and applied it to a medical ontology chunk. Our evaluation shows, that the strategy to derive propositions from a special corpus seems to end up in clearer and more error prone conceptual propositions to a domain expert. The experiments have to be repeated with other specialised corpora from the web, the major task for future work. Instead of Google one could refer to the hits of a medical search engine like Medivista.

From all our experiments we identified very important meta-properties of the algorithm. These are a possible basis for future extension of the algorithm: a more systematic treatment of the initial rule set ρ by gradually extending the word distances can be achieved, if we are able to keep the compression property of the algorithm.

From our point of view the integration of the presented algorithm in a Delphi method [9] for k-med-like projects or the evolving semantic web is very promising. In the context of the task of the k-med ontology, automatically identifying a whole group of symptoms is especially helpful for managing documents for case-based medical education.

A number of other interesting questions comes along with the approach. Among them are the following ones: how do we construct and balance a suitable corpus to learn from, which linguistic preprocessing is necessary or helpful, how does the approach scale for larger ontologies. The latter question is again closely related to our observations: the op-

timisation problems end up in an identification of a few relevant collocation features and the representation matrix can stand a pruning preprocessing.
Also the question of evaluating the results is interesting for related areas such as Kohonen maps for documents and word clustering algorithms [17].

References

1. Spark-Jones K.: *Readings in Information Retrieval*, Morgan Kaufmann, 1997
2. Stumme, G., Mädche A.: *FCA-Merge: A Bottom-Up Approach for Merging Ontologies*, Proceedings of the 17th International Joint Conference on Artificial Intelligence, Seattle, USA, August, 1-6, 2001, San Francisco/CA: Morgan Kaufmann, 2001
3. Sahlgren,M.: *Vector-Based Semantic Analysis: Representing Word Meanings Based on Random Labels*, Proceedings of the ESSLLI 2001 Workshop on Semantic Knowledge Acquisition and Categorisation, Helsinki, Finland, 2001
4. Faatz, A., Hoermann, S., Seeberg, C., Steinmetz, R.: *Conceptual Enrichment of Ontologies by means of a generic and configurable approach*, workshop notes of the ESSLLI 2001-workshop on semantic knowledge acquisition and categorisation, Helsinki, Finland, August 2001
5. Resnik, P.: *Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language*, Journal of Artificial Intelligence Research, **11**, 1999
6. Kullback, S. and Leibler, R.A.: *On Information and Sufficiency*, Annals of MAthematical Statistics **22**, 1951
7. Dagan I., Perreira F., Lee L.: S*imilarity-based Estimation of Word Co-occurence Probabilities*, Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, ACL'94, New Mexico State University, June 1994
8. Stefan Hörmann and Ralf Steinmetz: *Creating courses with learning object metadata*, Multimedia Systems, Springer, Berlin/Heidelberg, to appear 2002
9. Clyde W. Holsapple and K.D. Joshi: *A collaborative approach to ontology design*, Communications of the ACM, Vol. **45**, 2, February 2002
10. Alexa, M.: *Text type analysis with TATOE*. Storrer, A. & B. Harriehausen (eds.) (1998): Hypermedia für Lexikon und Grammatik. Gunter Narr Verlag, Tübingen.

[Asium]: D. Faure and C. Nedellec: *ASIUM: Learning subcategorization frames and restrictions of selection*, in Y. Kodratoff, editor, 10th Conference on Machine Learning (ECML 98) -- Workshop on Text Mining, Chemnitz, Germany, April 1998

11. ampl optimisation software, *http://www.ampl.com*
12. Institut für deutsche Sprache, *www.ids-mannheim.de*
13. the Google search engine, *www.google.de*
14. Andreas Faatz: *Enrichment Evaluation*, technical report TR-AF-01-02 at Darmstadt University of Technology
15. Heyer, G.; Läuter, M.;Quasthoff, U.; Wittig, Th.; Wolff, Chr.: *Learning Relations using Collocations*, Proceedings of the IJCAI Workshop on Ontology Learning, Seattle, USA, August 2001

16. David Yarowsky: *Word-Sense Disambiguation using Statistical Models of Roget's Categories Trained on Large Corpora*, Proceedings of COLING-92, Nantes, France, 1992

17. Lagus, K.: *Studying similarities in term usage with self-organizing maps.* Proceedings of NordTerm 2001, 13-16 June, Tuusula, Finland. pp. 34-45. ,2001

18. Gaël de Chalendar and Brigitte Grau. "*SVETLAN' or how to Classify Words using their Context*", proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2000, Juan-les-Pins, France, October 2000, pages 203-216 Rose Dieng and Olivier Corby (Eds.), Springer, 2000,

19. Bisson, G. and Nédellec, C. and Cañamero L.: *Designing clustering methods for ontology building - The Mo'K workbench*, in Staab, S. and Maedche, A. and Nedellec C., editors, Ontology Learning ECAI-2000 Workshop, Berlin, August 2000

20. Hahn author Hahn, U. and Schnattinger}, *Ontology Engineering via Text Understanding*, IFIP'98 --- Proceedings of the 15th World Computer Congress, Vienna/Budapest, 1998

21. Eneko Aguirre, Mikel Lersundi: *Extracción de relaciones léxico-semánticas a partir de palabras derivadas usando patrones de definición.* Procesamiento del Lenguaje Natural 27: 165-172 (2001)

22. Lagus, K.: *Text Mining with the WEBSOM*. Acta Polytechnica Scandinavica, Mathematics and Computing Series no. **110**, Espoo, Finland 2000

23. Teuvo Kohonen, *Self Organising Maps*, 3rd Edition, Springer Series in Information Sciences, Vol. **30**, Springer, Berlin

Using Ontologies to Discover Domain-Level Web Usage Profiles

Honghua (Kathy) Dai and Bamshad Mobasher
{hdai,mobasher}@cs.depaul.edu

School of Computer Science, Telecommunication, and Information Systems,
DePaul University, Chicago, Illinois, USA

Abstract. Usage patterns discovered through Web usage mining are effective in capturing item-to-item and user-to-user relationships and similarities at the level of user sessions. Without the benefit of deeper domain knowledge, such patterns provide little insight into the underlying reasons for which such items or users are grouped together. This can lead to a number of important shortcomings in personalization systems based on Web usage mining or collaborative filtering. For example, if a new item is recently added to the Web site, it is not likely that the pages associated with the item would be a part of any of the discovered patterns, and thus these pages cannot be recommended. Keyword-based content-filtering approaches have been used to enhance the effectiveness of collaborative filtering systems by focusing on content similarity among items or pages. These approaches, however, are incapable of capturing more complex relationships at a deeper semantic level based on different types of attributes associated with structured objects. This paper represents work-in-progress towards creating a general framework for using domain ontologies to automatically characterize usage profiles containing a set of structured Web objects. Our motivation is to use this framework in the context of Web personalization, going beyond page- or item-level constructs, and using the full semantic power of the underlying ontology.

1 Introduction and Problem Statement

The goal of Web usage mining [24] is to capture and model the behavioral patterns and profiles of users interacting with a Web site. The discovered patterns are usually represented as collections of pages that are frequently accessed by groups of users with common needs or interests. Such patterns can be used to better understand behavioral characteristics of visitor or user segments, improve the organization and structure of the site, and create a personalized experience for visitors by providing dynamic recommendations [25, 7, 3, 15, 9, 10, 22].

At a conceptual level, there may be many different kinds of objects within a given site that are accessible to users. At the physical level, these objects may be represented by one or more Web pages. For example, consider a site containing information about movies. This site may contain pages related to the movies themselves, actors appearing in the movies, directors, studios, etc.

Conceptually, each of these entities represents a different type of semantic object. During a visit to this site, a user may access several of these objects together during a session.

Given the session-based Web usage data, a variety of mining techniques can be used to discover patterns, including clustering, association-rule or sequential pattern discovery. For example, clustering of user sessions may result in the discovery of a group of similar sessions based on pages commonly accessed within those sessions. In order to find an aggregate representation of user interests captured by such a cluster, one might derive the cluster centroid containing a set (or a vector) of pages that are common among cluster elements. In [18, 17], we called these aggregate representations of session clusters *aggregate usage profiles* and used these profiles for Web personalization.

Specifically, each user session s can be viewed as an n-dimensional vector over the space of all pages, i.e.,

$$t = \langle w(p_1, s), w(p_2, s), \ldots, w(p_n, s) \rangle \,,$$

where $w(p_i, t)$ is a weight, in session s, associated with the page p_i. The weights can be binary representing the existence or non-existence of a page access in the session, or they can be some value representing the user's interest on the page (e.g., the page stay time). Applying data mining techniques, such as clustering, to this space may result in a set $CL = \{cl_1, cl_2, \ldots, cl_k\}$ of session clusters, where each cl_i is a subset of the set of sessions.

Given a session cluster cl, we can construct a usage profile pr_{cl} as a set of pageview-weight pairs by computing the centroid of cl:

$$pr_{cl} = \{\langle p, weight(p, pr_{cl})\rangle \mid weight(p, pr_{cl}) \geq \mu\}$$

where

- the significance weight, $weight(p, pr_{cl})$, of the page p within the usage profile pr_{cl} is given by
$$weight(p, pr_{cl}) = \frac{1}{|cl|} \cdot \sum_{s \in cl} w(p, s);$$
- $w(p, s)$ is the weight of page p in session $s \in cl$; and
- the threshold μ is used to focus only on those pages in the cluster that appear in a sufficient number of sessions in that cluster.

Each such profile, in turn, can be represented as a vector in the original n-dimensional space. The aggregate representation of common usage patterns as sets or vectors of pages makes such usage profiles quite useful for personalization and collaborative filtering [12]: given a new user, u who has accessed a set of pages, P_u, so far, we can measure the similarity of P_u to the discovered profiles, and recommend to the user those pages in matching profiles which have not yet been accessed by the user.

While in the above discussion we have focused on clustering as the primary data mining technique for the discovery of usage profiles, it should be noted that a variety of other techniques can also be used. For example, frequent itemsets discovered as part of association rule mining [1] on the usage data, would also lead to sets of items or pages representing usage profiles [16].

One problem with the above approach is that the profiles only capture common usage patterns at the page level. They do not reflect the underlying reasons why the group of users represented by the profile are interested in the accessed pages. This can lead to a number of important shortcomings in personalization systems based on Web usage mining or collaborative filtering. For example, if a new item is recently added to the Web site, it is not likely that the pages associated with the item would be a part of any of the discovered patterns (due to a lack of sufficient usage level data). Yet, a user who fits one of the profiles may indeed be interested in the new item, because the underlying domain characteristics of the item might correspond to those of items in one or more of the profiles.

In the movie site example mentioned above, consider the situation where a discovered usage profile may contain pages related to a number of movies many of which are from the same genre and have been directed by the same director. Using the above standard approach for personalization, pages appearing in this profile may be recommended to a user who has accessed some of the other pages in that profile. However, if a new movie is added to the site with similar properties, it will not appear in any profile until a sufficient number of users have accessed this movie together with other similar movies. This problem is often coined as the "new item problem" in collaborative filtering.

A common approach to resolving this problem has been to integrate content characteristics of pages with the usage patterns [6, 20, 21, 19]. Generally, in these approaches, keywords are extracted from the content on the Web site and are used to either index pages by content or classify pages into various content categories. In the context of personalization, this approach would allow the system to recommend pages to a user, not only based on a matching usage profile, but also (or alternatively) based on the content similarity of these pages to the pages user has already visited.

Keyword-based approaches, however, are incapable to capturing more complex relationships among objects at a deeper semantic level based on the inherent properties associated with these objects. In our movie site example, the above content-based filtering approach would allow the system to recommend other movies based on similarities in their textual descriptions or other content characterisitcs. But, the system would have considerable difficulty in recommending, for example, unrelated movies from the same genre, having the same main actors as those already accessed by the user, etc. To be able recommend different types of complex objects using their underlying properties and attributes, the system must be able to rely on the characterization of user segments and objects, not just based on keywords, but at a deeper semantic level using the domain ontologies for the objects.

This paper represents work-in-progress towards creating a general framework for using domain ontologies to automatically characterize usage profiles containing a set of structured Web objects. Our motivation is to use this framework in the context of Web personalization, going beyond page-level constructs, and using the full semantic power of the underlying ontology. This effort involves the following tasks:

1. Given a page in the Web site, we must extract domain-level structured objects as semantic entities contained within this page. This task involves the automatic extraction and classification of objects of different types into classes based on the underlying domain ontologies. Our goal is to create a representation for a usage profile discovered through Web usage mining process (e.g., through clustering or association rule mining), not simply as a set of pages, but as a set of structured objects embedded in those pages.
2. Given a set of structured objects representing a usage profile, we must create an aggregated representation as a set of pseudo objects each characterizing objects of different types occurring commonly across the user sessions. We call such a set of aggregate pseudo objects a *Domain-level Aggregate Profile.* Thus, a domain-level aggregate profile characterizes a collection of similar users based on the common properties of objects in the domain ontology that were accessed by these users.

We begin by providing a formal framework for the representation of the domain ontology and for creating aggregate representations of domain-level objects. We then discuss how the resulting aggregate profiles can be used in the context of personalization.

2 Representing Domain Ontologies for Web Objects

There has been much recent work in designing ontology languages to formally represent knowledge on the Web, such as *RDFS* [2] and *DAML+OIL* [13]. The ontology language DAML+OIL has been extended to include formal semantics and reasoning support by mapping to the description logic $\mathcal{SHOQ}(D)$ [14].

Generally speaking, in our current work we adopt the syntax and semantics of $\mathcal{SHOQ}(D)$ to represent domain ontologies. In $\mathcal{SHOQ}(D)$, the notion of *concrete datatype* is used to specify literal values and *individuals* which represent real objects in the domain ontology. Moreover, *concepts* can be viewed as sets of individuals, and *roles* are binary relations between a pair of concepts or between concepts and datatypes. The detailed formal definitions for concepts and roles are given in [11, 14]. Because our current work does not focus on reasoning tasks such as deciding subsumption and membership, we do not focus our discussion on these operations. The reasoning apparatus in $\mathcal{SHOQ}(D)$ can be used to provide more intelligent data mining services as part of our future work.

In $\mathcal{SHOQ}(D)$, a concept has the meaning of a set of individuals, which in our context are called “objects”. The notion of a concept is quite general and may encompass a heterogeneous set of objects with different properties (roles) and structures. In the present work, we are interested in deriving aggregate representations of a group of objects that have a homogenous concept structure (i.e., have similar properties and data types). For example, we may be interested in a group of movie objects, each of which has specific values for properties such as "year", "genre", "actors", etc. For the purpose of presentation, we call such a group of objects a *class*. Thus, in our framework, the notion of a class represents a restriction of the notion of a concept in $\mathcal{SHOQ}(D)$.

More specifically, our notion of class can be defined in the context of $\mathcal{SHOQ}(D)$ as follows.

Definition 1 A *class* $C \sqsubseteq I$ (I is the set of all individuals in the domain ontology) is a set of objects where there exists a set of roles R, such that, $\forall r \in R, D_r = \{v_2 \mid (v_1, v_2) \in r\}$, and $C \sqsubseteq \forall R.D_r$.

We call the roles that characterize the class C *attributes*. These attributes together define the internal properties of the objects in C or relationships with other concepts that involve the objects in C. And we denote the domain of values of the attribute r as D_r. Furthermore, because we are specifically interested in aggregating objects at the attribute level, we extend the notion of a role to include a domain-specific combination function and an ordering relation.

More formally, a class C is characterized by a finite set of attributes A_C, where each attribute $a \in A_C$ is defined as follows.

Definition 2 Let C be a class in the domain ontology. An *attribute* $a \in A_C$ is a 4-tuple, $a = \langle T_a, D_a, \psi_a, \preceq_a \rangle$, where

- T_a is the *type* for the values for the attribute a.
- D_a is the domain of the values for a;
- $\preceq_a$ is an ordering relation among the values in D_a; and
- ψ_a is a *combination function* for the attribute a.

The "type" of an attribute in the above definition may be a concrete datatype or it may be a set of objects (individuals) belonging to another class. Given a type T_a for an attribute a, we have $D_a \sqsubseteq dom(T_a)$.

In the context of data mining, comparing and aggregating values are essential tasks. Therefore, ordering relations among values are necessary properties for attributes. We associate an ordering relation $\preceq_a$ with elements in D_a for each attribute a. The ordering relation $\preceq_a$ can be null (if no ordering is specified in the domain of values), or it can define a partial or a total order among the domain values. For standard types such as values from a continuous range, we assume the usual ordering. In cases when an attribute a represents a concept hierarchy, the domain values of a are a set of labels, and $\preceq_a$ is a partial order representing the "is-a" relation.

Furthermore, we associate a data mining operator, called the *combination function* ψ_a, with each attribute a. The combination function ψ_a defines an aggregation operation among the corresponding attribute values of a set of objects belonging to the same class. This function is essentially a generalization of the "mean" or "average" function applied to corresponding dimension values of set of vectors when computing the centroid vector. In this context, we assume that the combination function is specified as part of the domain ontology for each attribute of a class. An interesting extension would be to automatically learn the combination function for each attribute based on a set of positive and negative examples.

Classes in the ontology define the structural and semantic properties of objects in the domain which are "instances" of that class. Specifically, each object o in the domain is also characterized by a set of attributes A_o corresponding to the attributes of a class in the ontology. In order to more precisely define the notion of an object as an instance of a class, we first define the notion of an instance of an attribute.

Definition 3 Given an attribute $a = \langle T_a, D_a, \psi_a, \preceq_a \rangle$ and an attribute $b = \langle T_b, D_b, \psi_b, \preceq_b \rangle$, b is an *instance* of a, if $D_b \subseteq D_a, T_b = T_a, \psi_b = \psi_a$, and $\preceq_b$ is a restriction of $\preceq_a$ to D_b. The attribute b is a *null instance* of a, if $D_b = \emptyset$.

Definition 4 Given a class C with attribute set $A_C = \{a_1^C, a_2^C, \ldots, a_n^C\}$, we say that an object o is *instance of class* C, if o has attributes $A_o = \{a_1^o, a_2^o, \ldots, a_n^o\}$ such that, a_i^o is a, possibly null, instance of a_i^C, for $1 \leq i \leq n$.

In the context of $\mathcal{SHOQ}(D)$, we can use the concept inclusion axiom $\{o\} \sqsubseteq C$ to denote the "instance-of" relation between o and C. In general, the domain ontology can be represented as a set of assertions on classes and attributes.

Based on the definitions of attribute and object instances, we can provide a more formal representation of the combination function ψ_a. Let c be a class and $\{o_1, o_2, \ldots, o_n\}$ a set of object instances of c. Let $a \in A_C$ be an attribute of class c. The combination function ψ_a can be represented by:

$$\psi_a(\{\langle a_{o_1}, w_1\rangle, \langle a_{o_2}, w_2\rangle, \ldots, \langle a_{o_n}, w_n\rangle\}) = \langle a_{agg}, w_{agg}\rangle,$$

where each a_{o_i} belonging to object o_i is an instance of the attribute a, and each w_i is a weight associated with the attribute instance a_{o_i} representing the significance of that attribute relative to the other instances. Furthermore, a_{agg} is a *pseudo instance* of a meaning that it is an instance of a which does not belong to a real object in the underlying domain. The weight w_{agg} of a_{agg} is a function of $w_1, w_2, \ldots, w_n$.

Given a set of object instances, $\{o_1, o_2, \ldots, o_n\}$, of class C, a *domain-level aggregate profile* for these instances is obtained by applying the combination function for each attribute in c to all of the corresponding attribute instances across all objects $o_1, o_2, \ldots, o_n$.

An Example

As an example, let us come back to the movie Web site discussed in the previous section. The Web site includes collections of pages containing information about movies, actors, directors, etc. A collection of pages describing a specific movies might include information such as the movie title, genre, starring actors, director, etc. An actor or director's information may include name, filmography (a set of

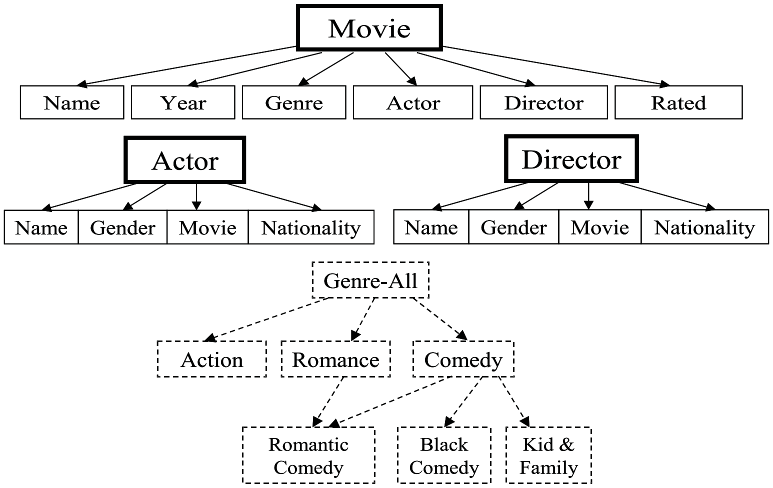


Fig. 1. The Ontology for a movie Web site

movies), gender, nationality, etc. The portion of domain ontology for this site, as described, contains the classes **Movie**, **Actor** and **Director** (see Figure 1). The collection of Web pages in the site represent a group of embedded objects that are the instances of these classes.

The class **Movie** has attributes *Year*, *Actor* (representing the role "acted by"), *Genre*, *Director*, etc. The *Actor*, and *Director* attributes have values that are other objects in the ontology, specifically, object instances of classes **Actor** and **Director**, respectively. The attribute *Year* is an example of an attribute whose datatype is positive integers with the usual ordering. The attribute *Genre* has a concrete datatype whose domain values in D_{Genre} are a set of labels (e.g., "Romance" and "Comedy"). The ordering relation for $\preceq_{Genre}$ defines a partial order based on the "is-a" relation among these labels (resulting in a concept hierarchy of Genre's a portion of which is shown in Figure 1).

An attribute a of an object o has a domain D_a. In cases when the attribute has unique value for an object, D_a is a singleton. For example, consider an object instance of class **Movie**, "About a Boy" (see Figure 2). The attribute *Actor* contains three objects (H. Grant, R. Weisz and T. Collette) that are instances of the class **Actor** (for the sake of presentation we use the Actor's name to stand for the object of Actor). Therefore, $D_{Actor} = \{$H. Grant, R. Weisz and T. Collette$\}$. Also, A real object may have values for only some of the attributes. In this case the other attributes have empty domains. For instance, the attribute *Director* in the example has an empty domain, and is thus not depicted in the figure.

We may, optionally, associate a weight with each value in the attribute domain D_a (usually in $[0, 1]$). This may be useful in capturing the relative importance of each attribute value. For example, in a given movie the main actors should have higher weights than other actors. In our example, the object actor *H.*

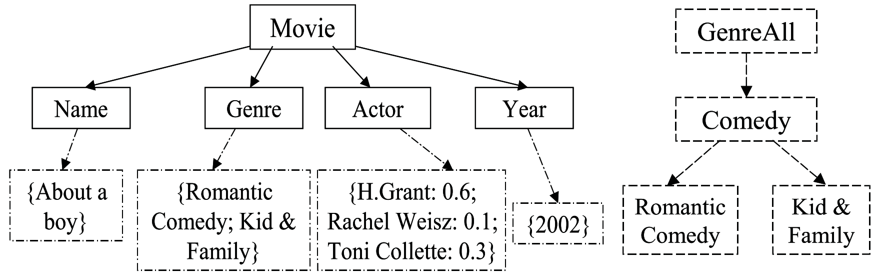

Movie
Name
Genre
Actor
Year
{About a boy}
{Romantic Comedy; Kid & Family}
{H.Grant: 0.6; Rachel Weisz: 0.1; Toni Collette: 0.3}
{2002}
GenreAll
Comedy
Romantic Comedy
Kid & Family

Fig. 2. An Example of an Object in Class Movie

Grant has weight 0.6 and the object Actor *Rachel Weisz* has weight 0.1. Unless otherwise specified, we assume that the weight associated with each attribute value is 1.

In the object o shown in Figure 2, the domain for the attribute *Genre* is the set {"Genre-All", "Comedy", "Romantic Comedy", "Kid & Family"}. The ordering relation $\preceq^o_{Genre}$ is a restriction of $\preceq_{Genre}$ to {Genre-All, Comedy, Romantic Comedy, Kid & Family}.

Let us now define the combination functions for some of the attributes in class **Movie**. Note that the combination functions are only applicable when creating an aggregate representation of a set of objects. For the attribute *Name*, we are interested in all the movie names appearing in the instances. Thus we can define ψ_{Name} to be the union operation performed on all the singleton *Name* attributes of all movie objects.

The attribute *Actor* contains a weighted set of objects in class **Actor**. In such cases we can use a vector-based weighted mean operation. The domain of the resulting aggregate attribute instance D'_{Actor} can be viewed as the union of the domains of the **Actor** attributes of the individual Movie objects: $D'_{Actor} = \cup_i D_{Actor_i}$, and the weight for an object o in D'_{Actor} is determined by

$$w'_o = \frac{\sum_i w_i \cdot w_o}{\sum_i w_i}.$$

For example, applying ψ_{Actor} to $\{\langle\{S, 0.7; T, 0.2; U, 0.1\}, 1\rangle, \langle\{S, 0.5; T, 0.5\}, 0.7\rangle, \langle\{W, 0.6; S, 0.4\}, 0.3\rangle\}$ will result in the aggregate domain D'_{Actor} of $\{\langle S, 0.58\rangle, \langle T, 0.27\rangle, \langle W, 0.09\rangle, \langle U, 0.05\rangle\}$.

As for the attribute *Year*, the combination function may create a range of all the *Year* values appearing in the objects. Another possible solution is to discretize the full *Year* range into decades and find the most common decades that are in the domains of the attribute. For example, applying ψ_{Year} to $\{\langle\{2002\}, 1>\}, \langle\{1999\}, 0.7\rangle, \langle\{2001\}, 0.3\rangle\}$ may result in an aggregate instance $Year'$ of attribute *Year* with $D'_{Year} = [1999, 2002]$.

The attribute *Genre* of class **Movie** contains a partial ordering relation $\preceq_{Genre}$ which represents a concept hierarchy among different *Genre* labels. Thus, for each instance object p of *Genre*, the relation $\preceq^a_{Genre}$ specifies the restriction of the partial ordering relation to D_p. The combination function, in this case, can perform tree (or graph) matching to extract the common parts of the conceptual hierarchies among all instances [23]. Given a set of objects $\{p_1, \ldots, p_n\}$ of *Genre*, we can define $\psi_{Genre}(\{\langle p_1, w_1\rangle, \langle p_2, w_2\rangle, \ldots, \langle p_n, w_n\rangle\}) = \langle \cap_i D_{p_i}, w'\rangle$, where w' is the average of the weights of instance values in the intersection. The relation $\preceq'_{Genre}$ is a restriction of $\preceq_{Genre}$ to $\cap_i D_{p_i}$. This function provides us the common segment of conceptual hierarchy for *Genre* in the set of movie instances. For example, applying ψ_{Genre} to {{"Romantic Comedy", "Kids&Family"}, {"Romance", "Comedy"}, {"Romance"}}, and assuming that all weights are 1, will result in an aggregate instance *Genre'* of attribute *Genre* with the value {"Romance"}.

3 Creating an Aggregated Representation of a Usage Profile

We now return to our original problem: how to create an aggregate representation of a discovered usage profile at the domain-level. As noted earlier, a usage profile at the page or item level can be viewed as a weighted set (or vector) of pages. The problem of extracting instances of the ontology classes from these pages is an interesting problem in its own right and has been studied extensively (see, for example, [5]). Here we assume that, either using manual rules, or through supervised learning methods, we can extract various object instances represented by the pages in the original page- or item-level usage profile. Thus, the usage profile can be transformed into a weighted set of objects: $pr = \{\langle o_1, w_{o_1}\rangle, \langle o_2, w_{o_2}\rangle, \ldots, \langle o_n, w_{o_n}\rangle\}$ in which each o_i is an object in the underlying domain ontology and w_i represents o_i's significance in the profile pr. The profile represents a set of objects accessed together frequently by a group of users (as determined through Web usage mining). Our goal is to create an aggregate representation of this weighted set of objects to characterize the common interests of the user segment captures by the profile at the domain level.

Given the representation of a profile pr as a weighted set of objects, the objects in pr may be instances of different classes $c_1, c_2, \ldots, c_k$ in the ontology. The process of creating a domain-level aggregate profile begins by partitioning pr into collections of objects with each collection containing all objects that are instances of a specified class (in other words, the process of classifying the object instances in pr). Let $g_i = \{\langle o_1^{c_i}, w_{o_1^{c_i}}\rangle, \ldots, \langle o_m^{c_i}, w_{o_m^{c_i}}\rangle\}$ be the elements of pr that are instances of the class c_i.

Having partitioned pr into k groups of homogeneous objects, $g_1, \ldots, g_k$, the problem is reduced to creating aggregate representation of each partition g_i. This task is accomplished with the help of the combination functions for each of the attributes of c_i some of whose object instances are contained in g_i. Once the representatives for every partition of objects are created, we assign a significance weight to each representative to mark the importance of this group of objects in the profile. In our current implementation the significance weight for each representative is computed as the sum of all the object weights in the partition.

```
Input: A weighted set of objects: $O = \{\langle o_1, w_1\rangle, \ldots, \langle o_n, w_n\rangle\}$
Output: The domain-level aggregate of $O$: $\{\langle o'_1, w'_1\rangle, \ldots, \langle o'_k, w'_k\rangle\}$
Main Procedure:
    $Result = \emptyset$
    Partition objects in $O$ into $g_1, g_2, \ldots, g_k$,
        such that $\forall g_i, \exists$ a class $c_{g_i}$, with objects in $g_i$ being instances of $c_{g_i}$
    For each $g_i (i = 1, \ldots, k)$ do
        Build a pseudo object $o'_i$ to be an instance of $c_{g_i}$:
            For each attribute $a$ in the attribute set of class $c_{g_i}$ do
                Let $a^{o'_i}$ be an instance of $a$ in the object $o'_i$:
                    $a^{o'_i} = \psi_a(g_i)$
            Endfor
        $Result = Result \cup \{o'_i\}$
        $w'_i = \sum_{o_k \in g_i} w_k$
    Endfor
    Return $Result$
```

Fig. 3. The Algorithm DPA for Creating and Aggregate Representation of a Weighted Set of Objects

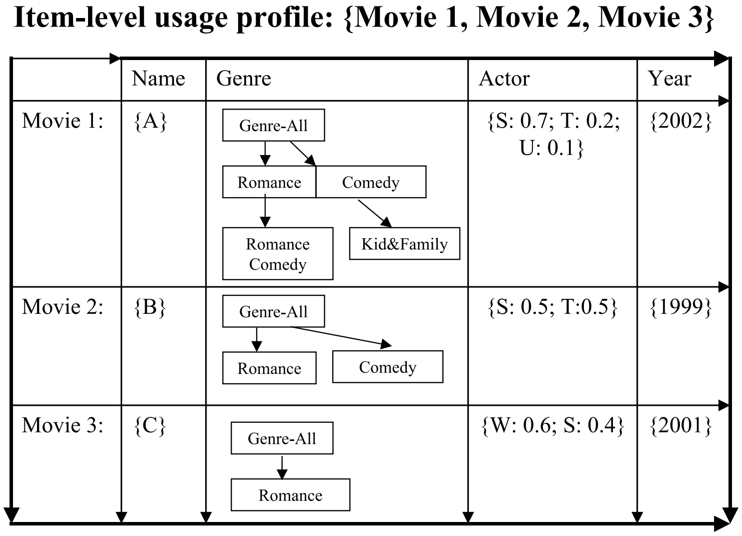
However, significance weight can be computed using other numeric aggregation functions. Figure 3 summarizes the algorithm DPA for creating domain-level aggregate profiles.

Examples Continued: Generating Domain-Level Aggregate Profiles

We present two examples of transforming usage profiles into domain-level aggregates using ontological information. The first example is based on our hypothetical movie Web site (Figure 1), and the second is based on the results of our experiments with a real Web site containing real estate property listings.

Suppose we have discovered a Web usage profile and transformed it into a weighted set of 3 movie objects in the class **Movie** (see Figure 4). We can generate a domain-level aggregate representation of these movies by applying the DPA algorithm to this profile. Because the objects are the instances of the same class, the DPA algorithm does not have to partition the objects. Thus, we can directly apply combination functions on each attribute in the class **Movie**. The details of combination functions used in this example were described in the previous section. Figure 5 shows the pseudo object instance representing the domain-level aggregation of these objects.

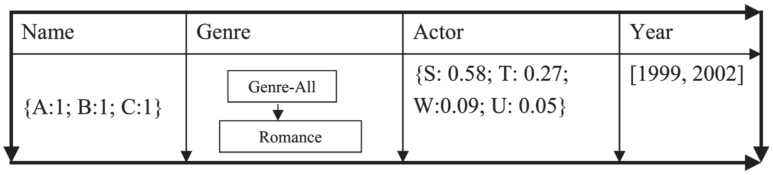
Note that the original item-level profile gives us little information about the reasons why these objects were commonly accesed together. However, after we characterize this profile at the domain-level, we find some interesting patterns: they all belong to *Genre* "Romance", and the actor S has a high score compared with other actors. This might tell us that this group of users are interested particularly in the movies belonging to "Romance" and are particularly fond of the actor S.



Item-level usage profile: {Movie 1, Movie 2, Movie 3}

	Name	Genre	Actor	Year
Movie 1:	{A}	Genre-All Romance Comedy Romance Comedy Kid&Family	{S: 0.7; T: 0.2; U: 0.1}	{2002}
Movie 2:	{B}	Genre-All Romance Comedy	{S: 0.5; T:0.5}	{1999}
Movie 3:	{C}	Genre-All Romance	{W: 0.6; S: 0.4}	{2001}

Fig. 4. A Weighted Set of Objects in a Usage Profile from a Movie Web Site



Name	Genre	Actor	Year
{A:1; B:1; C:1}	Genre-All Romance	{S: 0.58; T: 0.27; W:0.09; U: 0.05}	[1999, 2002]

Fig. 5. An Example of Domain-Level Aggregate Profile from a Movie Web site

Our next example is based on a real usage profile discovered from the usage data belonging to a real estate Web site containing various property listings. The ontology of the Web site has a single class **property**. An object which is the instance of class **property** represents a real estate property that has been listed for sale. This ontology is depicted in Figure 6.

In this case, all attributes have atomic values. For example, the attribute *number of bedrooms* has a range of values {1, 2, 3, 4, 5, 6} and *style of building* has values among {*1 Story*, *2 Story*, *Town Home*, *Ranch*}. Figure 6 also shows an example of a property object instance with the value sets for each attribute in dashed boxes.

In our experiment, we began by clustering similar user sessions. Then we generated the usage profiles obtaining the cluster centroids in which each item's weight represents the percentage of cluster sessions containing that item. For example, one of the usage profiles obtained from our experiment is: {⟨Property 1,1⟩,⟨Property 2,0.7⟩,⟨Property 3,0.18⟩,⟨Property 4,0.18⟩}.

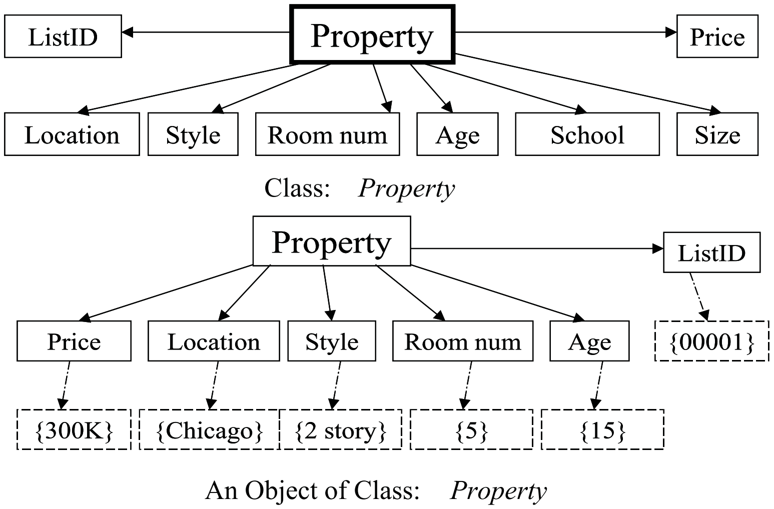


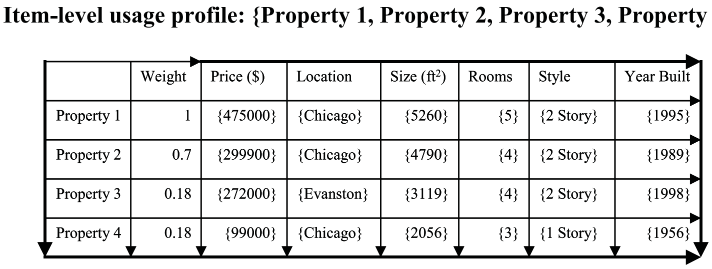
Fig. 6. The Ontology of a Real Estate Web site Containing Property Listings

We used the "weighted average" as our combination function for the attributes containing numeric data. Let w_{a_i} be the weight associated with instance a_i of attribute a. Because all the attribute instances contain singletons as their value sets, we can use v_{a_i} to represent the only value for the instance a_i . The combination function ψ_a for each of the numeric attributes (*Price*, *Size* and *Rooms*) gives the value set D'_a of the aggregate instance, as follows:

$$D'_a = \left\{ \frac{\sum_i v_{a_i} \times w_{a_i}}{\sum_i w_{a_i}} \right\}$$

For the attribute *Location*, the combination function computes the union of all the values in this attribute. In this example, this function returns {Chicago, Evanston}. For the attribute *Style*, the combination function computes the most frequent style values in this attribute. In this case, the function returns {2 Story}. Finally, we used the same combination function for the attribute *Year* as the one used in the class **movie** of the previous example. Figures 7 and 8 show the original usage profiles discovered through clustering and the resulting domain-level aggregate profile, respectively.

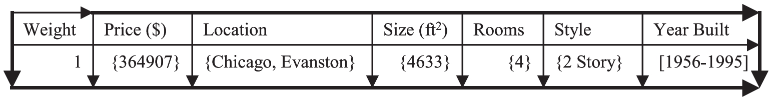
After generating domain-level usage profile, we have the information about the average price, size, number of rooms, as well as the locations, major styles, and year range. Such information not only enriches the profile with more knowledge about the listed items, but also provides us more possibilities for Web personalization. In the following section we discuss how we can leverage domain-level aggregate profiles for Web personalization.



Item-level usage profile: {Property 1, Property 2, Property 3, Property 4}

	Weight	Price ($)	Location	Size (ft^2)	Rooms	Style	Year Built
Property 1	1	{475000}	{Chicago}	{5260}	{5}	{2 Story}	{1995}
Property 2	0.7	{299900}	{Chicago}	{4790}	{4}	{2 Story}	{1989}
Property 3	0.18	{272000}	{Evanston}	{3119}	{4}	{2 Story}	{1998}
Property 4	0.18	{99000}	{Chicago}	{2056}	{3}	{1 Story}	{1956}

Fig. 7. A Weighted Set of Objects in a Usage Profile from a Real Estate Web Site



Weight	Price ($)	Location	Size (ft^2)	Rooms	Style	Year Built
1	{364907}	{Chicago, Evanston}	{4633}	{4}	{2 Story}	[1956-1995]

Fig. 8. An Example of Domain-Level Aggregate Profile from a Real Estate Web site

4 Web Personalization Based on Domain-Level Usage Profiles

Domain-level aggregate profiles present users' interests not just as a set of items or pages, but in terms of the common underlying properties and relationships among those items that are captured by object attributes. This fine-grained domain knowledge enables more powerful approaches to personalization.

We consider the browsing history of the current user to be a weighted set of items that the user has visited. The weight can be based on the amount of time the user spends on each item or other measures that represent the significance of the items in the current browsing session. The algorithm DPA can be applied to this weighted set to create an aggregate domain-level representation of the user browsing history. We call this aggregate representation the *current user model*. Given the current user model, there are two possible approaches to generating personalized recommendations for the user. These two approaches are depicted in Figure 9.

The first approach searches the domain ontologies to recommend the items that have similar features with the current user model. We call the recommendationed objects the *instantiations* of the current user model. This approach is essentially a generalization of the keyword-based content filtering approach and does not rely on any discovered usage profiles. While this approach does not encounter the "New Item" problem, it does not have the advantages of collaborative filtering systems, namely, considering item-to-item relationships that are based on how items are used or accessed together rather than based on content similarity. In addition, this approach requires the definition of possibly complex similarity or distance functions for each of the attributes of the underlying classes in the ontology.

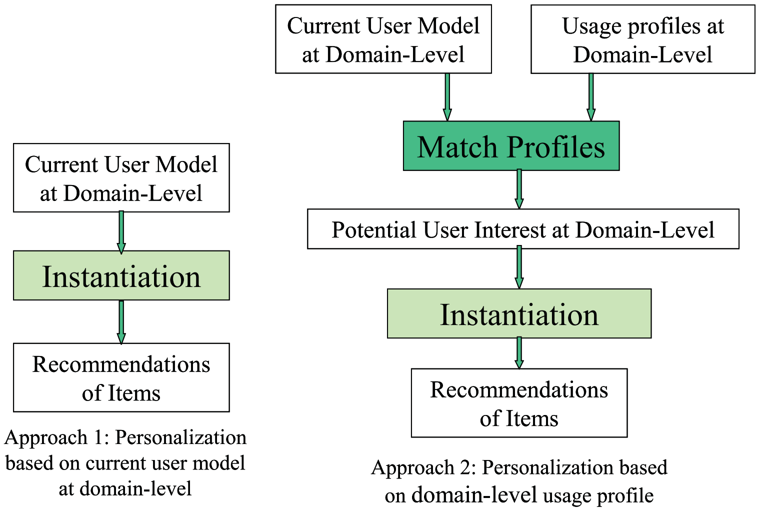

Current User Model at Domain-Level
Instantiation
Recommendations of Items
Approach 1: Personalization based on current user model at domain-level
Current User Model at Domain-Level
Usage profiles at Domain-Level
Match Profiles
Potential User Interest at Domain-Level
Instantiation
Recommendations of Items
Approach 2: Personalization based on domain-level usage profile

Fig. 9. Personalization Framework utilizing Domain Knowledge

The second method generates recommendations based on the discovered domain-level aggregate profiles. The recommendation engine matches the current user model with all the discovered usage profiles. The usage profiles with matching score greater than some pre-specified threshold are considered to represent this user's potential interests. A successful match implies that the current user shares common interests with the group of users this usage profile represents. The recommendation engine also matches the items in the domain with the user's potential interests and will recommend to the user the matching items. This approach is more complex than the pure content-based approach in that it involves two matching processes: matching the current user with the usage profiles and matching the item candidates with the user's potential interests.

The second approach has a number of advantages. First, it retains the user-to-user relationships that can be captured by the discovered usage profiles. Secondly, in contrast to standard collaborative filtering, it provides more flexibility in matching usage profiles with current user model because in this case matching involves comparison of features and relationships, not exact item identities. Furthermore, the items do not have to appear in any usage profiles in order to be recommended, since fine-grained domain relationships are considered during the instantiation process. The following example shows that this approach can also be used to solve the "New Item" problem.

Let us consider the real estate Web site discussed in the previous section. We again consider the real usage profile containing *property 1, property 2, property 3 and property 4* depicted in Figure 7. Suppose that there is a new item *property 5* (recently added to the site) which is not included in any usage profiles. Assume that *property 5* has attributes $\langle Price = \{370000\}, Location = \{Chicago\}, size = \{4500\}, rooms = \{4\}, style = \{2story\}, Yearbuilt = \{1993\}\rangle$.

Furthermore, assume that a user has browsed pages related to *property 1* and *property 2*. If the recommendation engine were based on item-level usage profiles, *property 5* would not be recommended. The recommendation engine would instead recommend *property 3, property 4*. However, the current user model (*property 1* and *property 2*) is indeed more similar to *property 5* than to *property 3* or *property 4* in terms of their domain-level attributes. If the recommendation engine uses domain-level aggregate profile of Figure 8, it would be able to recommend *property 5*.

5 Conclusion and Future Work

In this paper we have presented a general framework for using domain ontologies to automatically characterize usage profiles containing a set of structured Web objects. Our motivation has been to use this framework in the context of Web personalization, going beyond page- or item-level constrcuts, and using the full semantic power of the underlying ontology.

In our approach we consider a Web site as a collection of objects belonging to certain classes. Given a collection of similar user sessions (e.g., obtained through clustering) each containing a set of objects, we have shown how to create an aggregate representation of for the whole collection based on the attributes of each object as defined in the domain ontology. This aggregate representation is a set of pseudo objects each characterizing objects of different types commonly occurring across the user sessions. We have also presented a framework for Web personalization based on domain-level aggregate profiles.

Currently we assume that we have the predefined combination functions for each class attribute specified as part of the domain ontology. One area of future work involves the study of machine learning techniques in order to discover the best way to summarize the attribute automatically. Another area of future work involves the creation of general as well as domain-specific distance ro similarity functions allowing for the comparison of objects in different classes with the domain-level profiles. Finally another interesting area of work will be to explore use of discoverd domain-level aggregates from Web usage mining to enrich the existing domain ontology for a Web site.

References

1. R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. Proc. *20th Int. Conf. Very Large Data Bases, VLDB*, 1994.
2. D. Brickley, and R.V. Guha, Resource Description Framework (RDF) Schema Specification 1.0, *World Wide Web Consortium*, 2000. http://www.w3.org/TR/rdf-schema/
3. B. Berendt and M. Spiliopoulou. Analysing navigation behaviour in web sites integrating multiple information systems. In *VLDB Journal, Special Issue on Databases and the Web.* 9(1):56-75, 2000.

4. J. Broekstra, M. Klein, S. Decker, D. Fensel, F. van Harmelen, and I. Horrocks. Enabling knowledge representation on the web by extending RDF schema. In Proceedings of the 10th World Wide Web conference, Hong Kong, China, May 1–5, 2001.
5. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, S. Slattery. Learning to construct knowledge bases from the world wide web. In *Artificial Intelligence*, 118:69-113, 2000.
6. M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining Content-based and Collaborative Filters in an Online Newspaper. In *Proceedings of the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation.* University of California, Berkeley, Aug. 1999.
7. R. Cooley, B. Mobasher and J. Srivastava. Data preparation for mining World Wide Web browsing patterns. *Journal of Knowledge and Information Systems*, (1) 1, 1999.
8. M. Deshpande and G. Karypis. Selective markov models for predicting web-page accesses. In *First International SIAM Conference on Data Mining*, 2001.
9. E. Damiani, B. Oliboni, E. Quintarelli, and L. Tanca. Modeling users' navigation history. In *Workshop on Intelligent Techniques for Web Personalisation at the 17th International Joint Conference on Articial Intelligence (IJCAI01)*, Seattle, Washington, (USA), 2001.
10. X. Fu, J. Budzik, and K. J. Hammond. Mining navigation history for recommendation. In *Proc. 2000 International Conf. Intelligent User Interfaces*, New Orleans, LA, January 2000. ACM.
11. R. Giugno and T. Lukasiewicz. P-SHOQ(D): A Probabilistic Extension of SHOQ(D) for Probabilistic Ontologies in the Semantic Web. Accepted for publication in Proceedings of *the 8th European Conference on Logics in Artificial Intelligence (JELIA'02)*, Cosenza, Italy, September 2002. Lecture Notes in Artificial Intelligence, Springer, 2002.
12. J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of ACM SIGIR'99*, 1999.
13. Ian Horrocks. DAML+OIL: a reason-able web ontology language. In *Proc. of EDBT 2002*, March 2002. To appear.
14. I. Horrocks and U. Sattler. Ontology reasoning in the SHOQ(D) description logic. In B. Nebel, editor, *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI-01)*, pages 199-204. Morgan Kaufmann, 2001.
15. B. Mobasher, R. Cooley and J. Srivastava. Automatic personalization based on Web usage mining. In *Communications of the ACM*, (43) 8, August 2000.
16. B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Effective Personalization Based on Association Rule Discovery from Web Usage Data. *In Proceedings of the 3rd ACM Workshop on Web Information and Data Management (WIDM01), held in conjunction with the International Conference on Information and Knowledge Management (CIKM 2001)*, Atlanta, Georgia, November 2001.
17. B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Improving the effectiveness of Collaborative Filtering on Anonymous Web Usage Data. *In Proceedings of the IJCAI 2001 Workshop on Intelligent Techniques for Web Personalization (ITWP01)*, August 2001, Seattle.
18. B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Discovery of Aggregate Usage Profiles for Web Personalization. *In Proceedings of the Web Mining for E-Commerce Workshop (WebKDD'2000), held in conjunction with the ACM-SIGKDD Conference on Knowledge Discovery in Databases (KDD'2000)*, August 2000, Boston.

19. B. Mobasher, H. Dai, T. Luo, Y. Sun, and J. Zhu. Combining web usage and content mining for more effective personalization. In *Proceedings of the International Conference on ECommerce and Web Technologies (ECWeb)*, 2000.
20. D. Mladenic. Text learning and related intelligent agents: a survey. *IEEE Intelligent Systems*, 14(4):44–54, 1999.
21. M. Pazzani. A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review*, Dec. 1999, pp. 393-408.
22. Pitkow J. and Pirolli P. Mining Longest Repeating Subsequences to Predict WWW Surfing. In *Proceedings of the 1999 USENIX Annual Technical Conference*, 1999.
23. R. Ramesh, L. V. Ramakrishnan. Nonlinear pattern matching in trees. In *Journal of the ACM*, 39(2):295-316, 1992.
24. J. Srivastava, R. Cooley, M. Deshpande, P-T. Tan. Web usage mining: discovery and applications of usage patterns from Web data. *SIGKDD Explorations*, (1) 2, 2000.
25. Myra Spiliopoulou and Lukas C. Faulstich. WUM: A Tool for Web Utilization Analysis. In *Proceedings of EDBT Workshop WebDB'98*, Valencia, Spain, Mar. 1998.

Multi-faceted Learning for Web Taxonomies

Wray Buntine and Henry Tirri

Helsinki Inst. of Information Technology
HIIT, P.O. Box 9800
FIN-02015 HUT, Finland
{wray.buntine,henry.tirri}@hiit.fi

Abstract. A standard problem for internet commerce is the task of building a product taxonomy from web pages, without access to corporate databases. However, a nasty aspect of the real world is that most web-pages have multiple facets. A web page might contain information about both cameras and computers, as well as having both specification and sale data. We are interested in methods for supervised and unsupervised learning of multiple faceted models. Here we present results for multi-faceted clustering of bigram words data.

1 Introduction

A recognized problem for internet commerce is the task of building a product taxonomy from web pages, without access to corporate databases, and then populating a database with link information about service, repair, spare parts, reviews, product specifications, product family and company home pages, and purchase information from retailers. A key precursor for this task is the ability to build classification hierarchies in a supervised or unsupervised manner. However, a nasty aspect of the real world is that most web-pages have multiple facets. A web page might contain information about cameras and computers, as well as having both specification and sale data. Whereas another page might mix a product index with partial specification and sales data. We are interested in methods for supervised and unsupervised learning of multi-faceted models.

Clustering or unsupervised learning is now a standard method for analysing discrete data such as documents, and is now being used in industry to create taxonomies from web pages. A rich variety of methods exist borrowing theory and algorithms from a broad spectrum of computer science: spectral (eigenvector) methods [1], kd-trees [2], using existing high-performance graph partitioning algorithms from CAD [3], hierarchical algorithms [4] and data merging algorithms [5], etc.

All these methods, however, have one significant drawback for typical application in areas such as document or image analysis: each item/document is to be classified exclusively to one class. Their models make no allowance for instance, for a product page to have 60% digital camera content and 40% laptop computer content. It is 100% one way or another, and any uncertainty is only about whether to place the 100% into one or the other class. In practice documents invariable mix a few topics, readily seen by inspection of the human-classified Reuters newswire, so the automated construction of topic hierarchies needs to reflect this. One alternative is to make clustering *multi-faceted* whereby a document can be assigned proportionally (i.e., using a convex combination) across a number of clusters rather than uniquely to one cluster.

Authors have recently proposed discrete analogues to principle components analysis (PCA) intended to handle discrete or positive only count data of the kind used in the bag-of-words representation of web pages. Methods include non-negative matrix factorization [6], probabilistic latent semantic analysis [7] latent Dirichlet allocation [8], multinomial PCA [9]. A good discussion of the motivation for these techniques can be found in [7], and an analysis of related reduced dimension models and some of the earlier statistical literature here can be found in [10], and theory and algorithms are presented in [9].

Multinomial PCA by itself does not perform multi-faceted clustering because on average a page/item/document might be composed of up to 50 components in some of our experiments, and this does not reflect the behaviour of a "few different topics" we were looking for. We have recently made modifications to the standard algorithms so that multinomial PCA performs multi-faceted clustering. That is, it performs a clustering whereby some items/documents/pages are assigned with proportion to a few different classes.

In this paper, we first expand more on what we mean by a multi-faceted model, and then we give some examples from our working system. Our experiments are conducted on bigram data for words collected off a good fraction of Google's database of web pages for August 2001. The data sizes allowed us to experiment to understand how many components might be produced.

2 Contrasting Clustering and Multi-Faceted Clustering

For concreteness, consider the problem in terms of the usual "bag of words" representation for a document [11]. Here the items making up the sample are documents and the features are the counts of words in the document. A document is represented as a sparse vector of words and their occurrence counts. All positional information is lost. With J different words, the dimensionality for words/features, each document becomes a vector $\boldsymbol{x} \in \mathcal{Z}^J$, where the total $\sum_j x_j$ might be known. Traditional clustering becomes the problem of forming a mapping $\mathcal{Z}^J \mapsto \{1, \ldots, K\}$, where K is the number of clusters. Whereas techniques such as PCA form a mapping $\mathcal{Z}^J \mapsto \mathcal{R}^K$ where K is considerably less than J.

The problem we consider, however, is to represent the document as a convex combination, thus to form a mapping $\mathcal{Z}^I \mapsto \mathcal{C}^K$ where $\mathcal{C}^K$ denotes the subspace of $\mathcal{R}^K$ where every entry is non-negative and the entries sum to 1 ($\boldsymbol{m} \in \mathcal{C}^K$ implies $0 \leq m_k \leq 1$ and $\sum_k m_k = 1$). Call $\boldsymbol{m}$ the reduced image of a document.

For instance, suppose we are performing a coarse clustering of newswires into topics: the topics found might be "sports", "business", "travel", "international", "politics", "domestic", and "cultural". Consider a document about a major sport-star and the overlap of his honeymoon with a big game. Then traditional clustering might output the following: “the document is about sports”. A more refined clustering system that represents uncertainties as well might output: “with 90% probability it is about sports, with 7% probability it is about cultural, and 3% probability about something else”. General multinomial PCA considered in this paper might output: “50% of the document is about sports, 35% of the document is about cultural, 7% about business, 5% about international”. The supposed business content is really a discussion of the hotel for the honeymoon and the supposed international content comes from the location of the honeymoon. Note here general multinomial PCA plays the role of dimensionality reduction, and places similar kinds of words into the same bucket for compression purposes rather than any real topic identification.

The problem we consider is also to perform multi-faceted clustering, which serves the purpose of extracting multiple mutually occurring topics from a document. Suppose $\boldsymbol{m} \in \mathcal{C}^K$ is the reduced image of a particular document. For multi-faceted clustering, $\boldsymbol{m}$ should have most entries zero, and only a few entries significantly depart from zero. A measure we shall use for this is entropy, $H(\boldsymbol{m}) = \sum_j m_j \log(1/m_j)$. Thus multi-faceted clustering prefers low entropy reduced images from $\mathcal{C}^K$. In the limit, when the average entropy of the reduced images is 0, the mapping becomes equivalent to standard clustering. With the simple honeymoon example above, the output could be reduced to: “70% of the document is about sports, 30% of the document is about cultural”. This makes the document have $2^{H(\boldsymbol{m})} = 1.85$ effective topics, as opposed to the original PCA example above with more proportions (0.5,0.35,0.07,0.05) which had $2^{H(\boldsymbol{m})} = 3.17$ topics.

Note that clustering is usually varied using the dimensionality K, their number of components, not the nature of their decomposition, $H(\boldsymbol{m})$. Approaches such as the Information Bottleneck method [12] and hierarchical approaches in general yield clustering at different scales and do not relax the assumption of mutual exclusivity. We make the distinction here between the term *component* which is a derived feature discovered for dimensionality reduction, and a *facet*, which is similar but is intended instead to be a relaxation of a topic. Documents should have a few facets for good multi-faceted clustering but many components for effective dimensionality reduction.

3 Theory

We briefly review the theory of the multinomial version of PCA, and discuss the extensions. More details of the basic theory appear in [9].

Given a document, we first to sample a K-dimensional probability vector $\boldsymbol{m}$ that represents the proportional weighting of components, and then to mix it with a $K \times J$ matrix $\boldsymbol{\Omega}$ whose k-th row represents a word probability vector for the k-th component. For a document with a total count of L words in its bag-of-words representation $\boldsymbol{x}$, this is modelled as:

$$\boldsymbol{m} \sim Dirichlet(\boldsymbol{\alpha}) \ ,$$

$$\boldsymbol{x} \sim Multinomial(\boldsymbol{m\Omega}, L) \ ,$$

where $\boldsymbol{\alpha}$ is a vector of K-dimensional parameters to the Dirichlet. Thus, the mean of each entry x_j is a convex combination of a column of $\boldsymbol{\Omega}$, the probabilities for the j-th word for different components.

This probability model does not readily yield an algorithm. The proportion vector $\boldsymbol{m}$ is a hidden variable but it cannot be treated with the standard EM algorithm for hidden variables. However, if we can introduce a second hidden variable for each document $\boldsymbol{w}$ which is the word counts now broken out by word index j and topic index k as a $J \times K$ matrix. This matrix has row totals equal the bag of words data $\boldsymbol{x}$. An algorithm can be derived which iteratively recomputes an expected value for both the topic proportions $\boldsymbol{m}$ and the word counts broken out by topic $\boldsymbol{w}$.

The following iterative algorithm can be derived using variational methods [9].

Theorem 1. *Given the hidden variable model above, and the priors:* $\boldsymbol{m} \sim$ *Dirichlet*$(\boldsymbol{\alpha})$ *and* $\boldsymbol{\Omega}_{k_l,\cdot} \sim$ *Dirichlet*$(2\boldsymbol{f})$*, where* $\boldsymbol{f}$ *is an empirical word probability vector and* $\boldsymbol{\alpha}$ *is some other vector giving priors for the first Dirichlet. The following updates converge to a lower bound of* $\log p(\boldsymbol{x}, \boldsymbol{\Omega}, \boldsymbol{\alpha})$ *that is optimal for all product approximations* $q(\boldsymbol{m})q(\boldsymbol{w})$ *for the hidden value posterior* $p(\boldsymbol{m}, \boldsymbol{w} \,|\, \boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{x})$*. The subscript* $[i]$ *indicates values from the* i*-th document. For this the* K*-dimensional vector* $\boldsymbol{\beta}$ *is an intermediate variable representing a Dirichlet approximation to the posterior distribution for the* i*-th documents proportions* $\boldsymbol{m}_{[i]}$ *and the* $J \times K \times I$*-dimensional array* $\boldsymbol{\gamma}$ *is an intermediate variable representing the multinomial probabilities for an approximation to the posterior distribution for the rows of the* i*-th documents word matrix* $\boldsymbol{w}_{[i]}$*.*

$$\gamma_{j,k,[i]} \longleftarrow \frac{1}{Z_{1,j,[i]}} \Omega_{k,j} \exp\left(\Psi_0(\beta_{k,[i]}) - \Psi_0\left(\sum_k \beta_{k,[i]}\right)\right) \ ,$$

$$\beta_{k,[i]} \longleftarrow \alpha_k + \sum_j r_{j,[i]} \gamma_{j,k,[i]} \ ,$$

$$\Omega_{k,j} \longleftarrow \frac{1}{Z_{2,k}} \left(2f_j + \sum_i r_{j,[i]} \gamma_{j,k,[i]}\right) \ ,$$

where $\Psi_0()$ *is the digamma function, and* $Z_{1,j,[i]}$ *and* $Z_{2,k}$ *are some normalizing constants.*

The exponential in the first rewrite rule is an estimate of $m_{k,[i]}$) as $\exp(E_q\{\log m_{k,[i]})\})$ which tends to reduce the component entropy $H(\boldsymbol{m}_{[i]})$. Note the last rewrite rule is the standard MAP estimate for a multinomial parameter vector.

To reduces the entropies of the component proportions $\boldsymbol{m}_{[i]}$ even further, we use the additional updates:

$$\gamma_{j,k,[i]} \longleftarrow \frac{1}{Z_{1,j,[i]}} \Omega_{k,j} m_{k,[i]}$$

$$m_{k,[i]} \longleftarrow \frac{n_{k,[i]}}{\sum_k n_{k,[i]} - \lambda \left(H(\boldsymbol{m_{[i]}}) - \log 1/m_{k,[i]} \right)} ,$$

where λ is a Lagrange multiplier that can be increased to decrease the entropy $H(\boldsymbol{m}_{[i]})$. Note this replaces the above update for $\gamma_{j,k,[i]}$. These modified updates correspond to using an entropic prior $pr(\boldsymbol{m}_{[i]}) \propto \exp\left(-\lambda H(\boldsymbol{m}_{[i]})\right)$, which is a weighted version of Brand's [13].

4 Experimental Setup

Data was collected about word occurrences from a significant portion of the English language documents in Google's August 2001 crawl. After HTML and other tokens are removed, the basic text is processed to determine the most frequent 5000 words consisting only of letters 'a'-'z' ignoring case. Their co-occurrence data, the so-called bigram data was also collected. Bigrams are only counted for contiguous words in the same phrase: not broken by punctuation (excepting '-'), line breaks or other formatting tokens. The large number of documents used allows the bigram data to be 17% non-zero for bigrams of the top 5000 words. Note, some web pages contain seemingly random text and more than enough jargon. The top word "to" has $139,597,023$ occurrences and the $5,000$-th word "charity" has $920,343$ occurrences. The most frequent bigram is "to be" with $20,971,200$ occurrences, while the $1,000$-th most frequent is "included in" at $2,333,447$ occurrences.

In this case, the role of document in the theory is played by a word, and the role of word, is played by the words appearing after this word.

The code for our system is 1300 documented lines of C, with error checking, input parsing, diagnostic reporting, and component display. The code runs comparably to a PCA algorithm, converging in maybe 10-30 iterations, depending on the accuracy required. It outputs a HTML page with internal links representing the different aspects of the multi-faceted model constructed.

To measure the component entropies, we use $2^{H(\boldsymbol{m}\,|\,d)}$ where $H(\boldsymbol{m}\,|\,d)$ is the mean of the individual entropies $H(\boldsymbol{m}_{[i]})$ for each document (which is a conditional entropy, hence the notation).

5 Experimental Results

We conducted a number of experiments as listed below.

5.1 Basic Illustration

We clustered the 5000 most frequent words on the web into 1500 different multi-faceted classes based on their occurrence in bigrams. The average effective number of components per word (measured by $2^{\langle H(\boldsymbol{m}\,|\,d)\rangle}$) using standard multinomial PCA is about 30 and the distinctions are difficult to interpret in many cases. Using the modified version of multinomial PCA which reduces the entropy of the component vector $\boldsymbol{m}$, we got this to the more manageable figure of about 3 effective components per word. Many words had a majority component with probability over 0.7, while a few had up to 20 different components.

Below we give some examples of the different facets for different words. These are presented here to illustrate the method. Note the clusters here represent *word use* which is subtlety different to *word meaning.* Look at the examples for "wedding" below to see this. Our interpretation of each facet is given in italic prior to the list of words included in the facet,

"wedding": – *occasion:* birthday, christmas, romantic, holiday, holidays, vacation, wedding, anniversary, happy,;
- *jewelry:* rush, mini, gold, silver, jewelry, diamond, bell, wedding,

"love": – *affection:* kiss, love
- *preference:* prefer, expect, recommend, need, like, probably, suggest, love, want, mean, say, require, never, think
- *emotion:* confusion, pride, stress, pleasure, danger, fear, depression, honor, pain, comfort, britain, suffering, passion, joy, glory, concern, desire, wealth, beauty, strength, escape, feeling, insight, promise, satisfaction, peace, respect, love

"four": – *number:* seven, eight, five, nine, six, twelve, ten, four, three, twenty, two, one
- *some/few:* few, several, five, ten, six, four, couple, three, half

"scene": – *event:* universe, situation, era, meal, scene, incident, lesson, province, instance, issue, game, case, event, series, state, class, mission, project, school, sale, unit
- *play/performance:* festival, scene

"efforts": – *group work:* initiative, initiatives, projects, proposals, collaboration, efforts, effort, programs, activities, strategies, work
- *attempts:* attempts, aim, attempt, efforts, effort
- *relationships:* minds, hearts, lives, voices, bodies, families, efforts, commitment, attention, relationship, original, work

For instance, "love" is broken into an affection term, an preference term, and an emotion term, whereas "efforts" is broken into a group work term, an attempts term, and a relationship term.

5.2 Explaining Component Dimensionality

Why does standard multinomial PCA produce different effective number of components? We varied the input in a number of ways to explore this question.

First, we ran the system with different starting dimensions for the number of components allowed. Given I documents (i.e., words in the Google data) and J words/features per document (again, words in the Google data), then with K starting components, we are attempting to reduce the $I \times J$ word count matrix to the product of a $I \times K$ document topic matrix and a $K \times J$ topic to word mapping. Some of the K topics may be rarely used and contribute little, thus the effective number of total components can be much less. We measure this as $2^{H(\boldsymbol{p})}$ where $\boldsymbol{p}$ is the K-dimensional vector of mean proportion for a topic in all documents (i.e., the mean of the rows of the $I \times K$ document topic matrix). This contrasts with the effective number of topics/components per document $2^{H(\boldsymbol{m}\,|\,d)}$ which is the conditional entropy on the $I \times K$ document topic matrix, or the mean of the entropies of the rows of the $I \times K$ document topic matrix.

Second, we down-sampled the data. We sampled without replacement from the data vector to reduce the total size of each "document" by subsampling factors of 100, 1,000, 10,000, 100,000 respectively. Note the bigram word data had huge starting counts. This was done to produce "documents" of different sizes. The characteristics of the data sets produced are given in Figure 1. The X-axis is the subsampling factor. The solid line (axis on left) represents the proportion of non-zeros in the resultant data, and the dotted line (axis on right) represents the count of words in the resultant "document".

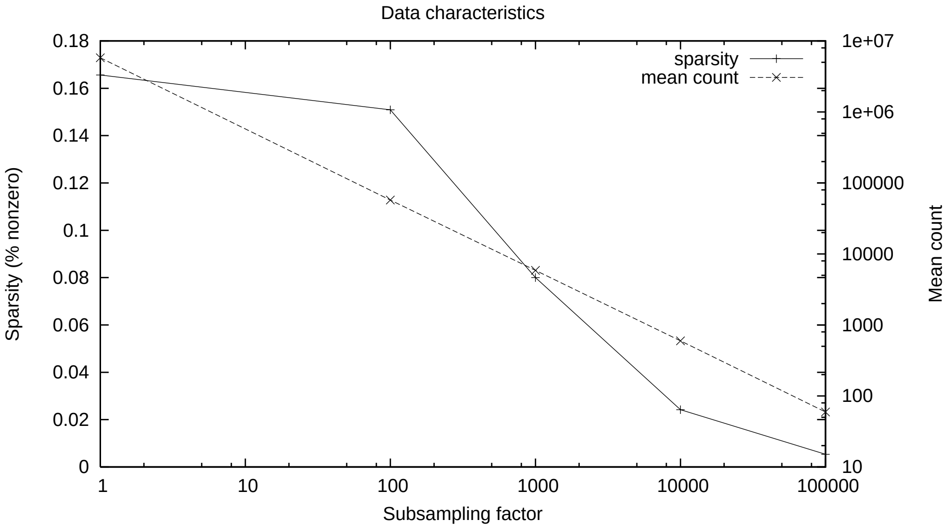


Fig. 1. Data characteristics for subsampled bigram data

The results from these experiments are reported in Figure 2. Each curve represents the results for one subsampling factor, i.e., an X value on Figure 1. So the top curve, which has no subsampling and where documents are 8,000,000 words on average, the mean effective number of components per document increases upto about 40. The second from the bottom curve (/10000) has about 80 words per document and typically 3-4 mean effective number of components per document, no matter how many components are inferred from the data (from 20 upto about 800). The third from the bottom curve (/1000) has about 800

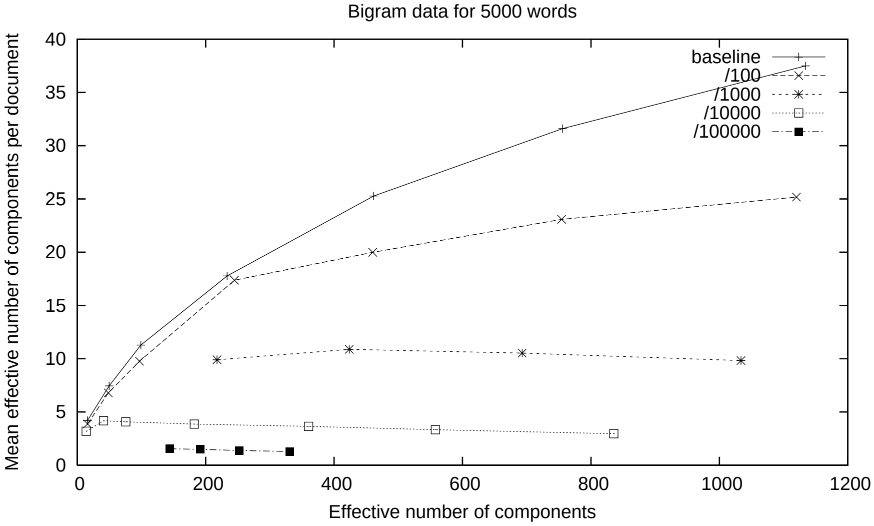


Fig. 2. Component dimensions

words per document and typically 10 mean effective number of components per document, no matter how many components are inferred from the data (from 20 upto about 1000).

From these experiments, we can conclude that the mean effective number of components is largely influenced by the document size. This would be result of the statistical capacity of the document to support a number of components. Small newswires and web pages can be 100 words, and thus could support a few topics statistically. Larger ones are about 1000 words and could support upto about 10 topics statistically. Web pages larger again that correspond to spec sheets, long details of factual data, etc., could support even more topics.

6 Conclusion

We have demonstrated that recent extensions to PCA for multinomial data are inadequate for multi-faceted clustering and given results for a modification to the basic algorithm that performs better in this regard. We argued that multinomial PCA is really a dimensionality reduction algorithm, and not designed for multi-faceted clustering.

It remains to be seen how the modified algorithm will perform on the suggested task of performing multi-faceted clustering of web-pages as a pre-processing step to data mining for the semantic web. For this, we would need at least the ability to generate full topic hierarchies, and to perform automatic labelling/naming of topics in the hierarchy. We expect that by doing this for multiple companies at once, useful hierarchies could be established.

Another research direction is to modify these algorithms to create supervised versions of them, whereby each item/document/page is tagged with multiple topics (as the Reuters and AP news-wires can be), and the task is to learn a model for the component topics and the proportions for each.

References

1. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 888–905
2. Moore, A.: Very fast EM-based mixture model clustering using multiresolution kd-tree. In: Neural Information Processing Systems, Denver (1998)
3. Han, E.H., Karypis, G., Kumar, V., Mobasher, B.: Clustering based on association rule hypergraphs. In: SIGMOD'97 Workshop on Research Issues on Data Mining and Knowledge. (1997)
4. Vaithyanathan, S., Dom, B.: Model-based hierarchical clustering. In: UAI-2000, Stanford (2000)
5. Bradley, P., Fayyad, U., Reina, C.: Scaling clustering algorithms to large databases. In: Proc. KDD'98. (1998)
6. Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. Nature **401** (1999) 788–791
7. Hofmann, T.: Probabilistic latent semantic indexing. In: Research and Development in Information Retrieval. (1999) 50–57
8. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. In: NIPS*14. (2002) to appear.
9. Buntine, W.L.: Variational extensions to EM and multinomial PCA. In: ECML 2002. (2002)
10. Hall, K., Hofmann, T.: Learning curved multinomial subfamilies for natural language processing and information retrieval. In: ICML 2000. (2000)
11. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley (1999)
12. Tishby, N., Pereira, F., Bialek, W.: The information bottleneck method. In: 37-th Annual Allerton Conference on Communication, Control and Computing. (1999) 368–377
13. Brand, M.: Structure learning in conditional probability models via an entropic prior and parameter extinction. Neural Computation **11** (1999) 1155–1182

Acquisition of Web Semantics based on a Multilingual Text-Mining Technique

Chung-Hong Lee[1] and Hsin-Chang Yang[2]

[1] Department of Electrical Engineering,
National Kaohsiung University of Applied Sciences,
415 Chien Kung Road, Kaohsiung 807, Taiwan
leechung@mail.cju.edu.tw
[2] Department of Information Management, Chang Jung University,
Tainan, Taiwan
hcyang@mail.cju.edu.tw

Abstract. This paper describes how semantic web issues such as machine-understandable semantic representation in multiple languages and acquisition of web semantics are tacked by means of a multilingual text-mining approach. This paper presents algorithms that enable multilingual text mining based on neural network technique, namely the self-organizing maps (SOM) for automatically grouping similar multilingual texts (i.e. Chinese and English texts). We treat the World Wide Web as our multilingual corpus and utilize the text-mining model for grouping semantically similar documents. When expose to raw multilingual text, the model clusters words and documents according to their semantic relatedness to form a semantic network.. The model also suggests a novel explanation of multilingual text clustering based on the semantic relatedness. In addition, we propose a framework based on the semantics extracted from the resulting document clusters, along with a XML document technique, to carry out the task of re-categorization of web pages. The web pages are re-categorized based on the semantic relatedness in the corpus. For implementation, we applied the derived multilingual text mining approach in the process of automatic construction of a new text directory, in order to a XML document database that achieves the aims of a semantics-based knowledge categorization.

1 Introduction

The "Semantic Web" is used to denote the next evolution step of the Web, which establishes a layer of machine understandable data. The data is suitable for automated agents, sophisticated search engines and interoperability services, which provide a previously not reachable grade of automation. The ultimate goal of the Semantic Web is to allow machines the sharing and exploitation of knowledge in the Web way, i.e. without central authority, with few basic rules, in a scalable, adaptable, extensible manner.

1.1 Semantics in Web Content

Recent work in computational linguistics suggests that large amounts of semantic information can be extracted automatically from large text corpora on the basis of lexical co-occurrence information. Such semantics has been becoming important and useful for being as an essential representation of content in each web page, particularly with the increasing availability of digital documents in various languages from all around the world. In this work we attempt to develop a novel algorithmic approach for multilingual text mining technique to extract semantic information from the web text corpora. Using a variation of automatic clustering techniques, which apply the *Self-Organizing Maps* (*SOM*), we have conducted several experiments to uncover associated documents based on Chinese-English bilingual parallel corpora, and a hybrid Chinese-English corpus. When exposed to raw multilingual text, the model clusters words and documents according to their semantic relatedness to form a semantic network, as a source of metadata for the construction of the Semantic Web.

2 Multilingual Text-Mining for Creating Metadata

Multilingual text mining is the extraction of implicit, previously unknown, and semantically similar terms and documents from multilingual corpora. The objective is to establish computer programs that automatically cluster conceptually related terms and texts in various languages.

In this work, the primary thrust is to make the text mining techniques fully multilingual, mining open-source texts in different languages. However our goal in these experiments is not to establish the ideal full-functional multilingual information discovery system, as the time and resources required for this task are considerable. Rather, we restrict our attention to bilingual corpora and try to understand the basic requirements for effective multilingual information discovery and the problems that arise from a simple implementation of such a system. This research work is mainly carried out by experimenting with text extraction from parallel corpora, a variation of bilingual corpora. Bilingual corpora can be used in many ways: For multilingual information access, bilingual corpora make it possible to investigate syntactic, semantic and lexical relationships between languages and are also important sources of contrastive evidence of language in usage. Generally speaking, there are two different types of bilingual corpora: parallel and comparable corpora. Parallel text corpora are sets of translation-equivalent texts, in which generally one text is the source text and the other(s) are translations. Comparable text corpora are sets of texts from pairs or multiples of languages which can be contrasted and compared because of their common features. However, unlike parallel (i.e. translation-equivalent) corpora, they always concern a restricted sublanguage. They offer a source of data on natural language lexical equivalents within a given domain [17]. For multilingual text mining, in this work we expect it to be acted as a starting point for exploring the impacts on linguistics issues with the machine learning approach to mining sensible linguistics elements from multilingual text collections. It is believed that, if the corpus is large enough, then statistical or other algorithm-based techniques can be used to produce bilingual term equivalents or associations by comparing which strings co-occur in the same sentences over the whole corpus. Therefore in this work we employed a parallel corpus, the bilingual magazine articles from the *Sinorama* Magazine, to create a set of dual-language training documents for discovering cross-language term associations and document associations.

3 SOM based Text-Mining Approach

3.1 Document Preprocessing

The proposed system focuses on the task of finding associations in collections of text. Based on association, similar documents, through the proposed text mining process, can be gathered in a cluster. For an English corpus, the document preprocessing is quite straightforward. Our approach begins with a standard practice in information retrieval (IR) [19] to encode documents with vectors, in which each component corresponds to a different word, and the value of the component reflects the frequency of word occurrence in the document. In practice, the resulting dimensionality of the space is often tremendously huge, since the number of dimensions is determined by the number of distinct indexed terms in the corpus. As a result, techniques for controlling the dimensionality of the vector space are required. Such a problem could be solved by eliminating some of the most common and some of the rarest words, and by applying a numerical algorithm such as *Latent Semantic Indexing (LSI)* [2] method. For a Chinese corpus, the document preprocessing is relatively complicated. Since a Chinese sentence is composed of characters without boundaries, segmentation is indispensable. We employ a dictionary, some morphological rules and an ambiguity resolution mechanism for segmentation. In addition, we also extract named organizations, people, and locations, along with date/time expressions and monetary and percentage expressions. The rest of the process is the same as that of text mining in an English corpus.

3.2 Self-Organizing Maps

The *self-organizing map* (*SOM*) [9][10] is one of the major unsupervised artificial neural network models. It basically provides a way for cluster analysis by producing a mapping of high dimensional input vectors onto a two-dimensional output space while preserving topological relations as faithfully as possible. After appropriate training iterations, the similar input items are grouped spatially close to one another. As such, the resulting map is capable of performing the clustering task in a completely unsupervised fashion. In this work we employ the SOM method to produce two maps for text mining, namely the *word cluster map* and the *document cluster map*.

3.3 The Word Cluster Map and Document Cluster Map

The word cluster map that is employed for document encoding is produced according to word similarities, measured by the similarity of the co-occurrence of the words. Conceptually related words tend to fall into the same or neighboring map nodes. By means of the SOM algorithm, word clusters can be ordered and organized as nodes on the map. Let $\mathbf{x}_i \in \Re^N$, $1 \le i \le M$, be the feature vector of the i^{th} document in the corpus, where N is the number of indexed terms and M is the number of documents. We used these vectors as the training inputs to the map. The map consists of a regular grid of processing units called *neurons*. Each neuron in the map has N synapses. Let $\mathbf{w}_j = \{w_{jn} | 1 \le n \le N\}$, $1 \le j \le J$, be the synaptic weight vector of the j^{th} neuron in the map, where J is the number of neurons on the map. We trained the map by the SOM algorithm:

Step 1. Randomly select a training vector $\mathbf{x}_i$ from the corpus.

Step 2. Find the neuron j with synaptic weights $\mathbf{w}_j$ which is closest to $\mathbf{x}_i$, i.e.

$$\left\| \mathbf{x}_i - \mathbf{w}_j \right\| = \min_k \left\| \mathbf{x}_i - \mathbf{w}_k \right\|. \tag{1}$$

Step 3. For every neuron l in the neighbor of node j, update its synaptic weights by

$$\mathbf{w}_l^{\text{new}} = \mathbf{w}_l^{\text{old}} + \alpha(t)(\mathbf{x}_i - \mathbf{w}_l^{\text{old}}), \tag{2}$$

where $\alpha(t)$ is the training gain at time stamp t.

Step 4. Increase time stamp t. If t reaches the preset maximum training time T, halt the training process; otherwise decrease $\alpha(t)$ and the neighborhood size, and go to Step 1.

The training process stops after time T which is sufficiently large that every feature vector may be selected as training input for certain times. The training gain and neighborhood size both decrease when t increases.

After the training process, the map forms a Word Cluster Map by labeling each neuron with certain words. For the n^{th} word in the corpus we construct an N-dimensional vector $\mathbf{v}_n$ in which only the n^{th} element is non-zero. To label the neurons, we present each $\mathbf{v}_n$ to the map and find the best matching neuron. Since the number of neurons is generally much smaller than the number of words, each neuron in the map may have multiple labels. We may say that a neuron forms a word cluster because the closely related words will map to the same neuron. The word cluster map autonomously clusters words according to their similarity of co-occurrence. Words that tend to be found in the same document will be mapped to close neurons in the map. For example, the Chinese words for "neural" and "network" often occur simultaneously in a document. They will map to the same neuron, or neighboring neurons, on the map. Words that do not occur in the same document will map to distant neurons on the map. Accordingly we can define the relationship between two words according to their corresponding neurons in the word cluster map, and the mining task will be performed based on such relationships. The trained map also forms a Document Cluster Map by labeling each neuron with certain documents. The document feature vectors $\mathbf{x}_i$ are presented to the map to label the neurons. Documents with similar keywords will map to the same or neighboring neurons. The similarity between two documents may be calculated by measuring the Euclidean distance between their mapped neurons in the map. Since the number of the neurons is much less than the number of the documents in the corpus, multiple documents may map to the same neuron. Thus a neuron forms a document cluster. Besides, neighboring neurons represent document clusters of similar meaning, i.e. high keyword co-occurrence frequency.

3.4 Discovery Algorithm

In this section we explain how the word cluster map and document cluster map effectively model the relationship between the words and documents. We transform a document to a vector of word occurrence. After the self-organizing process, two documents will map to near neurons if they contain similar word occurrences. When different words are labeled on the same neuron or near neurons on the word cluster map, they tend to occur in a restricted set of documents. On the other hand, if two words seldom co-occur in any document, they should not be labeled on near neurons. This is because the neuron may be viewed as representing a virtual document containing those words labeled on it. Two words will be mapped to the same neuron if, and only if, they often co-occur in the same document, otherwise the virtual document may not contain these words simultaneously. Neighboring neurons in the word cluster map represent word clusters containing similar words, i.e. words tend to co-occur in the same document. Hence the self-organizing map may measure the word co-occurrence similarity among documents. We define the similarity between the pth word and the qth word as follows:

$$D_1(p,q) = (1 + 2^{\|G(N_p)-G(N_q)\|})^{-1} \quad (3)$$

where Np is the neuron labeled by the pth word and $G(Np)$ is the two-dimensional grid location of Np. Such similarity measures the likelihood of the co-occurrence of words. Large similarity reveals that the two words often co-occur in the same set of documents, which may be considered as a kind of association pattern among the words.

On the document cluster map a neuron represents a document cluster that contains documents of similar meaning, which is defined by the set of highly co-occurring words they contain. Since we train the map using the encoded document feature vectors as input, the weight vector of a neuron represents the occurrence of the words in a virtual document. Such a virtual document may be used as the centroid of the document cluster associated with that neuron. On the document cluster map only those documents containing overlapping words may map to the same neuron. Documents containing non-overlapping words may map to distant neurons. Neighboring neurons represent document clusters with similar (overlapping) sets of words; thus the co-occurrence of words may be determined by the neighborhood spatially. For any document, we can find its similar documents by examining its mapped neuron and neighboring neurons in the document cluster map. The similarity between the *i*th and *k*th document is defined as follows:

$$D_2(i,k) = (1 + 2^{\|G(N_i) - G(N_k)\|})^{-1} \tag{4}$$

where *Ni* here is the neuron labeled by the *i*th document and *G*(*Ni*) is grid location of *Ni* as in Eq.3.

Documents which contain the same sets of words will definitely map to the same neuron, resulting in high similarity defined in Eq.4. Moreover, even if these documents do not contain exactly the same set of words, we may still say that they are conceptually similar because 1) they still contain common words that often co-occur in these documents, and 2) the dissimilar words are likely occurring in documents mapped to the nearby neurons. Document cluster maps provide an effectively way to form document clusters. A neuron on the map represents a document cluster. We can also define the similarity between two clusters by the distance of their corresponding neurons:

$$D_3(j,l) = (1 + 2^{\|G(N_j) - G(N_l)\|})^{-1} \tag{5}$$

where *j* and *l* are the neuron indices of the two clusters.

The similarities defined in Eq.3, 4, and 5 provide some knowledge about the documents in the corpus. We use Eq.3 to discriminate words based on the knowledge discovered from the corpus. This is also true for Eq.4 and Eq.5 to discriminate documents and clusters respectively. Word associations, as well as document associations, are clearly defined by such similarities. It is natural to apply such associations to applications of document retrieval, indexing, and clustering. A query, which is formulated as a keyword or a combination of keywords adjacent by Boolean operators, is sent by the user to retrieve relevant documents. We perform a search through the word cluster map to obtain the labeled neurons of the keyword or combinations of keywords. The documents labeled on the corresponding neurons on the document cluster map are selected and presented to the user. Moreover, all documents may also be presented by document similarity defined in Eq.4. Documents with higher similarities appear earlier in the query result. The user gets a list of relevant documents even when the query words may not occur in the documents. Important information and new knowledge related to the user's query may be revealed by our method.

4 Multilingual Text Mining from a Hybrid Chinese-English Corpus

Furthermore, we tested the developed discovery algorithm against articles of a hybrid Chinese-English corpus, generated by mixing the Chinese web-documents with the English web-documents from the selected bilingual magazine articles in the *Sinorama* Magazine. In the corpus, each bilingual document pair is split into two documents: an English and a Chinese document. In this case, each training document in either Chinese or English in the corpus is deconstructed into a bag of Chinese and English terms respectively. Although the document pairs contained in the parallel corpus are translation-equivalent, we regard each one of the document pair as an independent training document. The developed SOM method treats this document as a bag of freely intermingled Chinese and English terms. The Chinese-text part of the corpus was first preprocessed by the word segmentation program as in previous experiments. For English documents, the document preprocessing still begins with term indexing, establishing stop lists, stemming, and then encoding documents with binary vectors, in which each component corresponds to a different word, and the value of the component reflects the presence of word in the document.

We constructed self-organizing maps that contain 36 neurons in 6×6 grid format to conduct experiments based on a small collection of the corpora (58 Chinese articles and 58 English articles). The resulting maps give the impression that the effectiveness of text clustering is quite significant in a mixture of English and Chinese experimental articles. An example of resulting word clusters from the trained word cluster map is shown in Table 1.

sinorama 工作 光華 bad bridg caught childlik chingju chrissi commun comprehens contribut countryw cultiv curios drove easili endlessli event eventu fulfil goal greatest highest impart inexhaust inferior jiafong joi magazin mission model modesti plant potenti problem profession pursu record repres respons scholar sens serv spark specialist transmit wai wang wonder 人中 人生 不只 不亞於 不斷 之間 充當 本國 生態 目的 丟人 她們 成就 自然 似乎 我們 赤忱 使命感 其他 委員 孩子 後來 後進 既然 根基 留下 追求 做好 執著 培養 專家 帶動 啓發 深厚 現在 責任 這些 提到 傳遞 敬業樂群 楷模 態度 榮譽 撰述 潛力 稿子 學者 擔任 樹立 橋樑 環保 總是 總編輯 謙遜 職守 灌輸

Table 1. An example of resulting word clusters from the trained word cluster map.

5 System Framework

The system framework is shown in Fig. 1. By applying the SOM neural-network technique, we extract semantics from the contents of a huge HTML-document collection using the developed multilingual text-mining algorithm [12][13]. The extracted semantics can be used to produce metadata and ontology to describe the information in the original web documents. Therefore, the original web documents can be transformed into XML documents and stored in the XML document database with the developed ontology for text categorization. As such, the existing web pages can be analyzed to generate a set of metadata to describe their content and produce an ontology for the XML document database through a text-mining technique, based on their semantic relatedness.

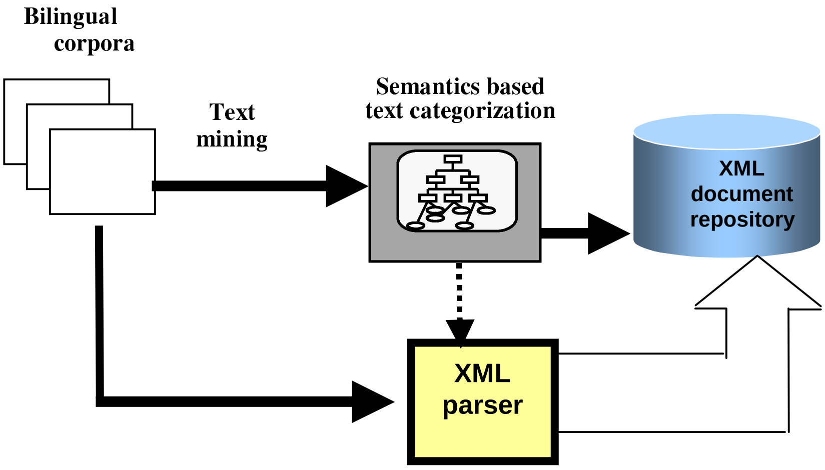


Fig. 1. System Framework

6 Related Work

Text mining is a new interdisciplinary field. It combines the disciplines of data mining, information extraction, information retrieval, text categorization, machine learning, and computational linguistics to discover structure, patterns, and knowledge in large textual corpora. With the huge amount of information available online, the World Wide Web has been becoming a fertile area for text mining research. The Web content data include unstructured data such as free texts, semi-structured data such as HTML documents, and a more structured data such as tabular data in the databases. However, much of the Web content data is unstructured text data. The research around applying knowledge discovery techniques to unstructured text is termed knowledge discovery in texts (KDT) [3], or text data mining [6], or text mining. Advances in computational resources and new statistical algorithms for text analysis have helped text mining develop as a field. Recently there have been some innovative techniques developed for text mining. For example, Feldman uses text category labels (associated with Reuters newswire) to find unexpected patterns among text articles [1][3][4][5]. Text mining by using *self-organizing map* (*SOM*) techniques has already gained some attentions in the knowledge discovery research and information retrieval field. The paper of [16] perhaps marks the first attempt to utilize SOM (unsupervised neural networks) for an information retrieval work. In this paper, however, the document representation is made from 25 manually selected indexed terms and is thus not really realistic. In addition, among the most influential work we certainly have to mention WEBSOM [7][8][11]. Their work aims at constructing methods for exploring full-text document collections, the WEBSOM started from Honkela's suggestion of using the *self-organizing semantic maps* [18] as a preprocessing stage for encoding documents. Such maps are, in turn, used to automatically organize (i.e., cluster) documents according to the words that they contain. When the documents are organized, following the steps in the preprocessing stage, on a map in such a way that nearby locations contain similar documents, exploration of the collection is facilitated by the intuitive neighborhood relations. Thus, users can easily navigate a word category map and zoom in on groups of documents related to a specific group of words. Differing from above traditional utilization of Self-Organizing Maps in text mining, in this project (including previous work [12][14][15][20][21]) we did not employ the map representation for visualization. Rather, we provide a concept-based query method for document retrieval from the database, in order to effectively improve the difficulties in navigating the maps for text search. Our system allows user to perform a keyword-based search similar to that of other search engines. Once user input a keyword, the input term was used as an entry point into the generated word cluster and the searching tool would return a list of suggested documents that were related to the input term, according to the SOM clustering algorithm. With particular emphases on processing multilingual information sources, the developed SOM technique performs a language-neutral method to tackle the linguistics difficulties in the text mining process. The algorithm was presented in detail in our previous work [12][14][15][20][21].

7 Conclusions

Still, in this case text mining from the "hybrid Chinese-English corpus" text collection actually represents typical similarity rather than translation equivalence. However, in this case terms including both Chinese and English ones which are conceptually similar are grouped in a neuron and neighboring neurons. In addition, an interesting outcome is that the resulting cluster groups words with similar concepts which are represented either in English or Chinese. This implies the developed mining algorithm is neutral to languages (i.e. English and Chinese) used in the experimenting documents. This potential provides possibilities to establish an effective way for concept extraction from multilingual information sources. Furthermore, the resulting term clusters can be used to establish a structure of concepts or an ontology. The ontology is often language-independent, or at least language-neutral, so that it can be used in multilingual Semantic Web applications.

Acknowledgement. This research has been partially supported by the National Science Council in Taiwan (NSC 90-2213-E-309-001).

References

1. Dagan, I., Feldman, R. and Hirsh, H."Keyword-Based Browsing and Analysis of Large Document Sets". Proc. of the Fifth Annual Symposium on Document Analysis and Information Retrieval (SDAIR), Las Vegas, NV, (1996).
2. Deerwester, S., Dumais, S., Furnas, G. and Landauer, K. "Indexing by Latent Semantic Analysis," Journal of American Society for Information Science 40(6), (1990) 391-407.

3. Feldman, R. and Dagan, I. "KDT - Knowledge Discovery in Texts". Proc. of the First Annual Conference on Knowledge Discovery and Data Mining (KDD), Montreal (1995).
4. Feldman, R., Klosgen, W. and Zilberstein, A. "Visualization Techniques to Explore Data Mining Results for Document Collections". Proc. of the Third Annual Conference on Knowledge Discovery and Data Mining (KDD), Newport Beach (1997).
5. Feldman, R., Dagan, I. and Hirsh, H. "Mining Text Using Keyword Distributions". J. of Intelligent Information Systems, Vol. 10, (1998) 281-300.
6. Hearst, M.A. "Untangling Text Data Mining". Proceedings of ACL'99: the 37th Annual Meeting of Association for Computational Linguistics, (1999).
7. Honkela, T., Kaski, S., Lagus, K. and Kohonen, T. "Newsgroup Exploration with WEBSOM Method and Browsing Interface". Technical Report A32. Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland (1996).
8. Kaski, S., Honkela, T., Lagus, K. and Kohonen, T. "WEBSOM--Self-Organizing Maps of Document Collections. Neurocomputing", Vol. 21 (1998) 101-117.
9. Kohonen, T. "Self-Organizing Formation of Topologically Correct Feature Maps," Biological Cybernetics, Vol. 43, pp.59-69, (1982).
10. Kohonen, T. Self-Organizing Maps, Springer-Verlag, Berlin, 1995.
11. Kohonen, T. "Self-Organization of Very Large Document Collections: State of the Art". In Niklasson, L., Boden, M., and Ziemke, T., editors, Proc. of ICANN98, the 8th International Conference on Artificial Neural Networks, Vol. 1, London, (1998) 65-74. Springer.
12. Lee, C.H. and Yang, H.C., "Towards Multilingual Information Discovery through a SOM based Text Mining Approach". In Proceedings of International Workshop on Text and Web Mining, The Sixth Pacific Rim International Conference on Artificial Intelligence (PRICAI 2000), Melbourne, Australia, August 28-September 1, (2000) 81-87.
13. Lee, C.H. and Yang, H.C., "A Multilingual Text Mining Approach Based on Self-Organizing Maps". In *Applied Intelligence* (The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies): Special Issue on Text and Web Mining. (SCI)(EI). In Press. (2003).
14. Lee, C.H. and Yang, H.C., "A Web Text Mining Approach Based on Self-Organizing Map". Proc. of the ACM CIKM'99 2nd Workshop on Web Information and Data Management (WIDM'99), Kansas City, Missouri, USA, (1999) 59-62.
15. Lee, C.H. and Yang, H.C., "A Text Data Mining Approach Using a Chinese Corpus Based on Self-Organizing Map". Proc. of the Fourth International Workshop on Information Retrieval with Asian Language (IRAL'99), Taipei, Taiwan, (1999) 19-22.
16. Lin, X., Soergel, D. and Marchionini, G. "A Self-Organizing Semantic Map for Information Retrieval". Proc. of the ACM SIGIR Int'l Conf on Research and Development in Information Retrieval (SIGIR'91), Chicago, IL (1991).
17. Picchi, E. and Peters, C. "Cross-Language Information Retrieval: A System for Comparable Corpus Querying". In Cross-Language Information Retrieval. eds. by Grefenstette, G. Kluwer Academic Publishers, (1998) 81-92.
18. Ritter, H. and Kohonen, T. "Self-Organizing Semantic Maps". Biological Cybernetics, Vol. 61 (1989) 241-254.
19. Salton, G. and McGill, M.J. Introduction to Modern Information Retrieval, McGraw-Hill Book Company, New York, 1983.
20. Yang, H.C. and Lee, C.H. (2000), "Automatic Category Generation for Text Documents by Self-organizing Maps". In Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN 2000), Como, Italy, July 24-27, 2000. Vol. III-581-586.
21. Yang, H.C. and Lee, C.H. (2000), "Automatic Category Structure Generation and Categorization of Chinese Text Documents". In The Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000), Lyon, France, Sep. 13-16, 2000.

An Empirical Investigation of Learning from the Semantic Web

Peter Edwards, Gunnar AAstrand Grimnes[*], and Alun Preece

Department of Computing Science
King's College
University of Aberdeen
Aberdeen, AB24 5UE
Scotland
+44 (0)1224 272270
{pedwards,ggrimnes,apreece}@csd.abdn.ac.uk

Abstract. The Semantic Web is a vision of a machine readable Web of resources, interlinked and connected through meta-data with common ontologies. In this paper we explore the impact such a Semantic Web would have on Machine Learning algorithms used for user profiling and personalisation. Our hypothesis is that learning from the Semantic Web should outperform traditional learning from today's World Wide Web for both performance and accuracy. In this paper we present results obtained with two different datasets marked-up with semantic meta-data; using these we have investigated different instance representations and various learning techniques. Our initial results with the Naïve Bayes and K-NN algorithms were disappointing, leading us to examine the use of the Progol algorithm. Using ILP techniques we were able to discover meaningful and we believe, potentially reusable knowledge.

[*] Author to whom all correspondence should be addressed.

1 Introduction

The Semantic Web [2][1] is a vision in which today's Web will be extended with machine readable content, and where every resource will be marked-up using machine readable meta-data. The intention is that documents on the Semantic Web will convey real meaning by using structured data-formats and by referring to common ontologies. We believe that initially the Semantic Web will consist of hand-crafted pages much like the Web we know today, providing the same information, but in machine readable form. For example, we could envisage semantic markup accompanying a conventional HTML page giving information such as: *This is the web page of Gunnar Grimnes, he works for Aberdeen University, his telephone number is 1224 630538 etc.* We believe that such static information demonstrates only part of the potential for Semantic Web technologies; their deployment should allow for advanced profiling methods capable of acquiring knowledge such as: *Gunnar likes bands who recorded most of their material from 1968 to 1975, as well as any band who uses a Moog Synthesiser.* When a true Semantic Web exists in this form it becomes useful, and should, for instance, allow for better matching of semantically enriched product descriptions with semantic user profiles.

Machine learning technologies have been applied in the context of today's World Wide Web to help users find their way through the unmanageable amount of information that exists. A typical scenario involves acquisition of a model of a user's interests which can then be used to make recommendations, e.g. *this link should be of interest* or *consider this product, it is similar to things you've bought previously* . A variety of approaches exist for learning from Web-content; these range from methods which choose to ignore all HTML markup and treat everything as plain text, to those which make use of the limited structure in HTML and treat the title, heading, link-texts differently. Once content has been extracted from documents, the next step is to apply information retrieval techniques, such as stopword removal, stemming, term weighting and so on. A bag-of-words representation is then used to form the training instances required by the learning algorithm. Figure 1 provides a schematic view of this process.

In the work presented here we wish to explore the impact of the Semantic Web on user profiling and personalisation; more specifically, we have investigated how machine-learning techniques could be used if we had access to semantic markup for every Web resource. Our hypothesis is that the Semantic Web should help solve fundamental problems that make machine learning from the Web today difficult, by providing structured information, reducing ambiguity, and providing useful references to background information in the form of ontologies. We suggest that learning from semantically marked-up data should outperform learning from unstructured or semi-structured text, with regard to increased accuracy, i.e. more meaningful and more usable results, as well as a decrease in the time and resources needed to execute the learning algorithm.

[1] World Wide Web Consortium Semantic Web Initiative, http://www.w3.org/2001/sw/

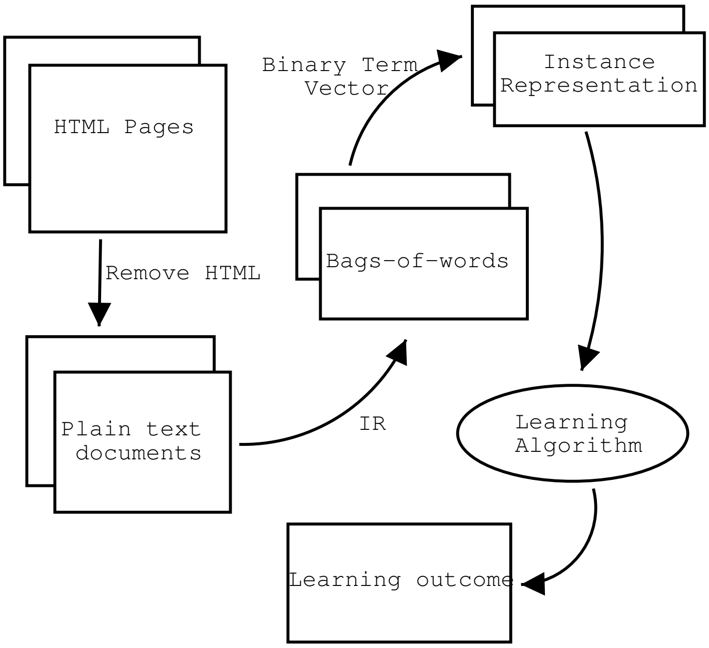


Fig. 1. Learning from the Web – Schematic View

1.1 Methodology

The approach we have taken is based on a typical machine learning application in the context of the World Wide Web: user-profiling. This scenario provides an opportunity to explore the behaviour of a number of learning algorithms. We assume that a user has interacted with a system on several occasions, perhaps by rating a Web page or making a purchase at an e-commerce site. For our purposes the exact scenario does not matter. Each of these interactions form a training instance, labelled with some class depending on the action performed by the user. For example (s)he might have rated a book "Very good", so the data about that book then becomes the instance and its class would be "Very good". The challenge is to use a set of such instances to acquire a classification model which can be used to predict class labels for future instances. For example, in an e-business context, such a model could then be used to recommend products to a user. To explore the impact of semantic markup, we require a number of datasets of interactions which exist in a semi-structured text format, as well as in a Semantic Web language, such as RDF[2]. We would then be able to compare the performance of learning from the plain text format with learning from semantic meta-data.

[2] http://www.w3.org/RDF/

In the next section, we describe the datasets we have used for our experiments, then in section 3 we discuss our experiments with knowledge sparse learning, the algorithms used, instance representations and results. In section 4 we discuss knowledge intensive learning, using the Inductive Logic Programming algorithm Progol, again discussing algorithms, document representation and results. Finally, we discuss some related work and present our conclusions.

2 Datasets

When commencing this work we knew that the Semantic Web was still very much in its infancy, but we still hoped that it would be possible to find semantically marked-up data upon which to base our experiments. Unfortunately, we have been unable to find any substantial amount of data containing such markup[3]. For this reason we were forced to generate our own (perhaps rather artificial) data. It is our hope, however, that this data will serve to illustrate the issues associated with learning from the Semantic Web.

2.1 The ITTalks Dataset

ITTalks[4] is an online portal for information about information technology seminars given at universities in the US. It was the only application of a Semantic Markup language we were able to find which had sufficient amounts of such data publicly available. It uses DAML+OIL[5] to describe talks, talkers, location and other concepts relating to seminars. The system is public and anyone is free to submit their own talk. The talks come in several formats, either as a plain HTML Web page or as DAML+OIL. Figure 2 shows an example of both formats.

Although these documents are generated from the same back-end database, there are some differences in content, e.g. the HTML version includes a biography of the author, while the DAML+OIL variant includes more information on time and place, etc. The ITTalks set contained descriptions of 64 talks, each of which formed an instance in our dataset. The 64 instances were manually classified by each of us (PE, GAG, ADP) into two classes, either *interesting* or *not interesting* based on the title, author and abstract. Three classified variants of the raw data were thus created. Figure 5 summarises the class distributions of each version.

2.2 The Citeseer Dataset

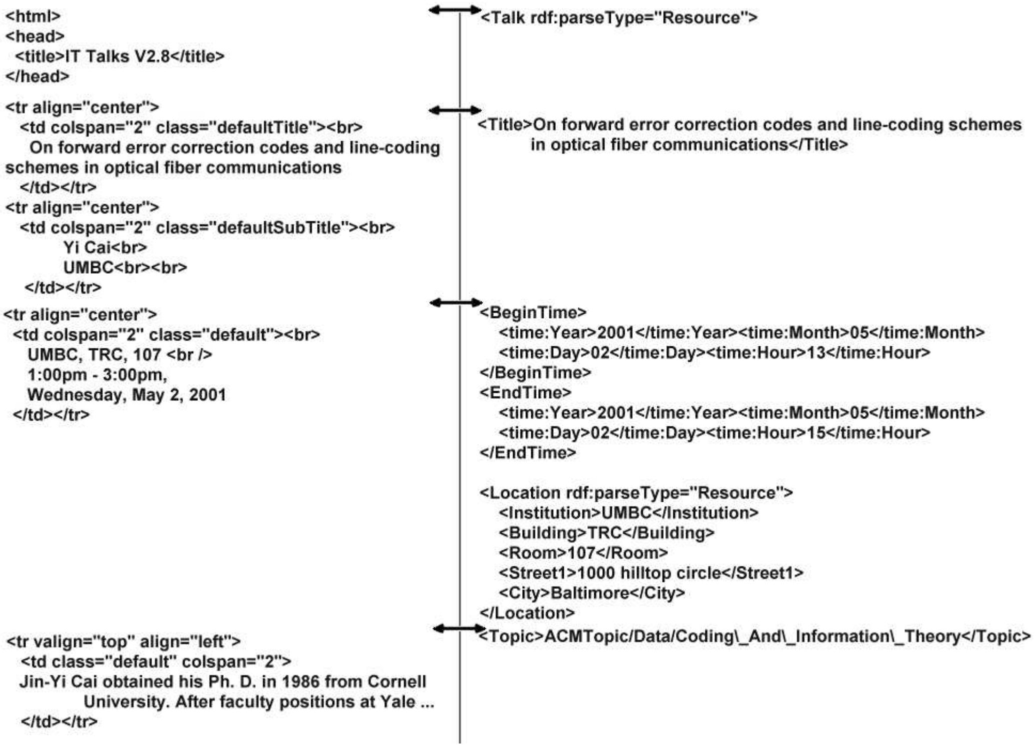
The NEC ResearchIndex[6] is a digital library of research papers within Computing Science. It does not provide documents with semantic markup, but the lack of any other sources of data with appropriate markup (other than ITTalks) forced us to look for different ways of acquiring such data. The ResearchIndex

[3] We would be delighted if anyone could point us at such a datasource!

[4] http://www.ittalks.org

[5] http://www.daml.org

[6] http://citeseer.nj.nec.com/cs



```
<html>
<head>
 <title>IT Talks V2.8</title>
</head>

<tr align="center">
  <td colspan="2" class="defaultTitle"><br>
   On forward error correction codes and line-coding
schemes in optical fiber communications
  </td></tr>
<tr align="center">
  <td colspan="2" class="defaultSubTitle"><br>
        Yi Cai<br>
        UMBC<br><br>
  </td></tr>
<tr align="center">
 <td colspan="2" class="default"><br>
   UMBC, TRC, 107 <br />
   1:00pm - 3:00pm,
   Wednesday, May 2, 2001
 </td></tr>

<tr valign="top" align="left">
  <td class="default" colspan="2">
  Jin-Yi Cai obtained his Ph. D. in 1986 from Cornell
          University. After faculty positions at Yale ...
  </td></tr>
```

```
<Talk rdf:parseType="Resource">

<Title>On forward error correction codes and line-coding schemes
        in optical fiber communications</Title>

<BeginTime>
   <time:Year>2001</time:Year><time:Month>05</time:Month>
   <time:Day>02</time:Day><time:Hour>13</time:Hour>
</BeginTime>
<EndTime>
   <time:Year>2001</time:Year><time:Month>05</time:Month>
   <time:Day>02</time:Day><time:Hour>15</time:Hour>
</EndTime>

<Location rdf:parseType="Resource">
   <Institution>UMBC</Institution>
   <Building>TRC</Building>
   <Room>107</Room>
   <Street1>1000 hilltop circle</Street1>
   <City>Baltimore</City>
</Location>
<Topic>ACMTopic/Data/Coding\_And\_Information\_Theory</Topic>
```

Fig. 2. HTML and DAML+OIL Examples from ITTalks.

provides the full text of the papers, and an indexing system for citations; in addition, it provides the corresponding BibTex entries, see Figure 3.

```
@inproceedings{ zucker92performance,
    author = "R. Zucker and J.-L. Baer",
    title = "A Performance Study of Memory Consistency Models",
    booktitle = "Proceedings of the 19th International Symposium on Computer Architecture",
    address = "Gold Coast, Australia",
    year = "1992",
    url = "citeseer.nj.nec.com/zucker92performance.html" }
```

Fig. 3. Example BibTex Entry from NEC ResearchIndex.

BibTex is a highly structured format, and is therefore easily converted to an XML based format, such as RDF. We chose RDF, as it is the W3C's basic Semantic Web representation language. The conversion to RDF was performed by making the identifier of the paper (i.e. *zucker92performance*) the subject and each attribute-value line a predicate and object in an RDF triple. Figure 4 illustrates the RDF generated from the BibTex appearing in Figure 3.

```
<?xml version="1.0"?>
<rdf:RDF
        xmlns="http://www.csd.abdn.ac.uk/~ggrimnes/exp/\#"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns\#">
<inproceedings rdf:about="zucker92performance">
        <author>R. Zucker and J.-L. Baer</author>
        <title>A Performance Study of Memory Consistency Models</title>
        <booktitle>Proceedings of the 19th International Symposium on Computer Architecture</booktitle>
        <address>Gold Coast, Australia</address>
        <year>1992</year>
        <url>citeseer.nj.nec.com/zucker92performance.html</url>
</inproceedings>
</rdf:RDF>
```

Fig. 4. RDF Example Generated from BibTex.

The meta-data generated from BibTex is very simple, and lacks many of the properties of a real Semantic Web resource, such as ontology references. One notable characteristic is that it is much shorter than the plain text representation of the same instance (typically ≈5000 words). In addition, the BibTex does include some information that is not to be found in the full text, such as *journal title* and *year of publication*.

The ResearchIndex provides a *Computer Science Directory*[7] listing the most cited papers in each of 17 subject categories. In these categories ResearchIndex lists 5066 papers. However, a number of these appear under more than one category and were removed, to leave 4220 instances in our dataset. One approach

[7] http://citeseer.nj.nec.com/directory.html

to learning from this data would be to attempt to learn a classifier capable of discriminating between the 17 categories. However, as a multi-class problem is substantially harder than a simple binary like/dislike classification, we also ran the experiment attempting a binary classification over each of the 17 categories, i.e. *is this paper an* Agents *paper, or is it any of the other 16 classes?* We chose the five categories with the most instances for binary classification, namely *Machine Learning, Artificial Intelligence, Information Retrieval, Human Computer Interaction* and *Databases.* We also selected the *Agents* as one of the smaller groups. Figure 6 presents the class distribution for this dataset. Note that the GAG variant has 10 more instances as more talks became available from ITTalks, however these were never classified by PE or ADP.

	Instances	**Likes**	**Dislikes**
GAG	64	25	39
PE	53	16	37
ADP	53	15	38

Fig. 5. ITTalks Dataset Profile.

	Dataset	**ML**	**AI**	**IR**	**HCI**	**DB**	**Agents**
Number of Instances	4220	385	367	337	284	285	218

Fig. 6. ResearchIndex Dataset Profile.

3 Knowledge Sparse Learning

3.1 Introduction

We define *Knowledge Sparse Learning* as the approach that is traditionally taken by statistical or probabilistic machine learning methods. In the Web context, the presence or absence of certain discriminating keywords within a set of training instances is used to produce a classification model. Such a model is then used to predict the class of future unseen instances. Such a model typically consists of a set of weights or a probabilistic model for terms occurring within each class. A model learned from such an approach is mainly useful within the given experiment, and will be useless for classifying instances from a different subject domain, where the discriminating keywords might be different, or even instances from the same domain which have been preprocessed in a different way.

3.2 Algorithms

For this part of our study, we used two well known machine learning algorithms, both of which operate using a feature vector representation of each instance. The features used to describe instances varied between our different approaches (see below). N-fold cross validation [21] was used when assessing the performance of the methods. The algorithms will now be discussed briefly.

Naïve Bayes [11] is a simplified version of the Bayes classifier, which takes a probabilistic approach to learning. Naïve Bayes reduces the complexity of the normal Bayes classifier by making the assumption that the features of an instance are conditionally independent. In practice, this assumption seldom holds true, but the algorithm still performs well on many real-world classification tasks, having been shown to be equal in accuracy to neural networks and decision tree learning [10].

K-Nearest Neighbour (KNN) [4] is an algorithm which will label unknown instances with the label of the majority of the K nearest neighbours in N-dimensional space, where N is the number of features used for describing each instance. KNN is a lazy algorithm, meaning that it will not generate a model based on the training instances, but only when asked to classify a new instance will it perform the computations needed to classify the given instance. The version of KNN used in our experiments is a variation of the standard algorithm, able to deal with symbolic inputs [3].

3.3 Instance Representations

We have investigated three different ways of representing instances in this study: one based on the text content of our documents, and two which make use (in some way) of the semantic meta-data. We will now describe each of these approaches.

1. Plain Text As a baseline approach we used a simple method to create training data [7, 1]. This was based upon the HTML version of the ITTalks data and the full text of the ResearchIndex articles. For each instance we removed all numbers and all words of length less than 3, before applying a stopword list which removes non-content words such as *it, the, their, etc.* We also explored application of a stemming algorithm [18], reducing words like *computer, computing, computers* to *comput.* The idea was to make generalising over classes easier, but we found stemming to make little or no difference in performance, and chose not to use it. Once this initial pre-processing had been completed, we had the option of either creating a binary vector with each element corresponding to a term in the document vocabulary, or some form of weighted vector (based on a subset of available terms). Due to the size of the vocabulary (150,000 terms for the ResearchIndex dataset) we decided to adopt the latter approach.

TF/IDF weights [6] were calculated for each of the terms and the 1500 with the highest TF/IDF ranking were selected. The presence or absence of each of these terms was then used to create a binary term vector. Figure 7 shows an example of the processing stages involved and the final representation.

Original HTML document:

```
<html>
<head><title>Machine Learning from the Semantic Web</title></head>
<body>
<h1>Machine Learning from the Semantic Web</h1>
<i>By Gunnar AAstrand Grimnes</i>
<p>In this seminar we give details on our recent experiments on learning from the semantic web

...
```

⇓

Removal of HTML markup, stopwords and numbers:

machine learning semantic web machine learning semantic web gunnar aastrand grimnes seminar give details recent experiments learning semantic web ...

⇓

Selection of most discriminating terms using TF/IDF:
learning, semantic, ontology, agent, talk, experiments, url ...

Binary term vector for this instance:
1, 1, 0, 0, 0, 1, 0 ...

Fig. 7. Instance Representation – Method 1.

2. Treating RDF Tags as Plain Text Our next approach was similar to the first, but instead of employing a plain text representation for each instance we make use of the marked-up data. This data was preprocessed in essentially the same manner as the plain text, with one important difference. In HTML documents the tags provide formatting information, i.e. <*b*> tells us that this text should be printed in bold, while in the meta-data files the different XML/RDF tags represent some information about the meaning of the content, i.e. <*location*> tells us that a talk has a location. We did not want to ignore this information, so in this approach we treat the XML-tags as additional text content. As we use TF/IDF to select highly ranked terms to appear in the instance representation, commonly occurring tags will of of course be ignored. However, tags which occur infrequently will still find their way into the instance. Figure 8 shows an example of this instance representation.

3. Using RDF with Tag ⇒ Feature Mapping Our third approach was based on the observation that on the Semantic Web, markup provides structure, so instead of throwing away this structure by treating all the text as one unit we processed the content of each tag separately. Each element in our instance vector then became the set of words which occurred within a certain tag; the content

Original RDF document:

```
<xml>
  <rdf>
    <talk id='mlsemweb1'>
      <title>Machine Learning from the Semantic Web</title>
      <speaker>
         <name>Gunnar AAstrand Grimnes</name>
         <url>http://www.csd.abdn.ac.uk/~ggrimnes</url>
         <faxnumber>+44 1224 273422</faxnumber>
      </speaker>
...
```

⇓

Removal of stopwords and numbers:
xml rdf talk title machine learning semantic web speaker name gunnar aastrand grimnes name url csd abdn ggrimnes url faxnumber speaker ...

⇓

Selection of most discriminating terms using TF/IDF:

learning, semantic, ontology, faxnumber, agent, experiment ...

Binary term vector for this instance:

1, 1, 0, 1, 0, 0, ...

Fig. 8. Instance Representation – Method 2. (Note difference from term vector in Figure 7)

of this tag was pre-processed in the same manner as for methods 1 and 2, i.e. applying a stopword list, ignoring short words, etc. The length of the instance vector then became the number of unique tags in the documents, not the number of unique words. This approach is possible because the variants of both Naïve Bayes and K-Nearest Neighbour we used supported sets of values as elements in the instance vectors.

While this approach made sense for most tags, some tags had a clearly defined internal structure where information would be lost if pre-processed. An example is the *ACMTopic* field of the ITTalks dataset, which gives the topic of the talk as a string such as *ACMTopic/Computer_Systems_Organization/-Computer_Communication_Networks/Internetworking.* If preprocessed normally this would be broken up into separate terms and the accurate meaning lost, so we did not preprocess this tag. Figure 9 shows an example of this instance representation.

3.4 Results

The ITTalks results are shown in Figures 10 and 11. From these results, it is immediately noticeable that the GAG variant of the classified ITTalks data led to poor results. We believe that this is an artifact due to the manual classification of the data; the GAG variant reflects a less clearly defined interest profile than

Original RDF document:

```
<xml>
  <rdf>
    <talk id='mlsemweb1'>
      <title>Machine Learning from the Semantic Web</title>
      <speaker>
         <name>Gunnar AAstrand Grimnes</name>
         <url>http://www.csd.abdn.ac.uk/~ggrimnes</url>
      </speaker>

...
```

⇓

Removal of stopwords, numbers, etc. from tag content:

```
<xml>
  <rdf>
    <talk>
      <title>machine learning semantic web</title>
      <speaker>
         <name>gunnar aastrand grimnes</name>
         <url>csd abdn ggrimnes</url>
      </speaker>

...
```

⇓

Using the following tags as features:
talk, title, speaker, name, url ...
Instance:

{}, { *machine, learning, semantic, web* }, {}, {*gunnar, aastrand, grimnes*}, {*csd, abdn, ggrimnes*}
...

Fig. 9. Instance Representation – Method 3.

	GAG	PE	ADP	Average
1. Plain Text	48.27%	69.38%	65.30%	60.98%
2. RDF Tags as text	50.00%	65.30%	67.34%	60.88%
3. RDF Tags as Features	34.48%	40.81%	40.81%	38.70%

Fig. 10. Results for Naïve Bayes - ITTalks Dataset.

	GAG	PE	ADP	Average
Method 1: Plain Text	55.17%	73.47%	65.31%	64.65%
Method 2: RDF Tags as Text	56.90%	69.39%	59.18%	61.82%
Method 3: RDF Tags as Features	50.00%	65.31%	57.14%	57.48%

Fig. 11. Results for K-Nearest Neighbour - ITTalks Dataset.

	Multi Class	ML	AI	IR	HCI	DB	Agents
Method 1: Plain Tex	43.38%	83.84%	70.02%	77.35%	78.69%	85.02%	78.93%
Method 2: RDF Tags as Text	47.53%	91.71%	89.76%	91.06%	93.67%	94.43%	95.23%
Method 3: RDF Tags as Features	51.13%	89.62%	88.05%	90.33%	91.51%	91.85%	92.82%

Fig. 12. Results for Naïve Bayes - ResearchIndex Dataset.

	Multi Class	ML	AI	IR	HCI	DB	Agents
Method 1: Plain Text	46.52%	93.39%	91.26%	94.00%	93.96%	95.47%	96.40%
Method 2: RDF Tags as Text	26.47%	91.73%	90.73%	92.39%	93.41%	94.00%	94.95%
Method 3. RDF Tags as Features	24.19%	89.83%	89.69%	90.21%	93.10%	92.65%	98.58%

Fig. 13. Results for K-Nearest Neighbour - ResearchIndex Dataset.

the two other variants, both of which were created by academics researchers with specific interests. Another phenomenon we can observe from the results is the poor performance of Method 3 (mapping RDF tags to features). We believe this is caused by the large number of distinct tags which appear in the ITTalks meta-data, and the fact that the majority of the textual content is contained within very few tags, such as *Abstract, Bio-Sketch* and *Title*. This would cause the instance representation for Method 3 to have very sparse vectors with a few features containing large numbers of terms; there are thus many redundant features which do not provide any information that can be used for creating the model. Method 2 is not affected by this as the entire document is treated as one unit of text for instance generation purposes. Finally, we observe that Method 1 and Method 2 lead to very similar results. We believe that this is caused by another artifact of the dataset. All of the instances in the dataset contain the same set of DAML+OIL tags, even if these might be empty for certain instances. As we use TF/IDF to select highly ranked terms for inclusion in the instance vector, and tag names appear in all examples, they will never get into the final instance representation. Thus, the plain text and meta-data versions of this dataset are essentially the same. We would anticipate that in a true Semantic Web dataset, the meta-data would be richer in nature and its presence would provide more information than the pure text instances.

The ResearchIndex results can be found in Figure 12 and 13. The first column gives results for the multi-category problem, and as expected, they are poor. K-Nearest neighbour performs much worse than Naïve Bayes at the multi-class classification, we believe this is caused by K-nearest neighbour being very susceptible to the inclusion of irrelevant or redundant attributes, as the distance metric combine measurements for all of the features [16].

4 Knowledge Intensive Learning

4.1 Introduction

On the Semantic Web content is represented via a logical language in which *meaning* is clearly defined. In our first group of experiments, while some use was made of the structure provided by markup, its logical nature was ignored. By exploiting the full potential of the Semantic Web we argue that it should be possible to learn rules and statements in a logical representation that is similar to that used for the content.

4.2 Inductive Logic Programming

We have chosen to use the Progol Inductive Logic Programming (ILP) system. Progol has been defined as "A standard Prolog interpreter with inductive capabilities" [15]. It is able to learn knowledge (expressed as Prolog predicates) from supplied example instances and supporting background information. The algorithm has been successfully used in experiments for analysis of mutagenic activity amongst nitroaromatic molecules [20], drug design [5] and protein shape prediction [14]. The theory behind ILP and the original Progol algorithm is described in [13]. For our experiments we used CProgol4.4.[8]

4.3 Methodology

We explored the application of Progol in the context of the NEC ResearchIndex Dataset, as it is the larger of our datasets and maps easily to a Prolog representation. The ITTalks dataset has a much richer set of meta-data which is more difficult to represent in Prolog. In our RDF→Prolog mapping, we have used the simple RDF data model of {*subject, predicate, object*} triples; the ITTalks dataset is encoded using DAML+OIL, which, when treated as plain RDF, generates many complex reification triples which would shroud the meaning of the documents, compared to the intuitive meaning of the simpler triples from the ResearchIndex, such as *the author of this article is Gunnar Grimnes.*

Progol was run on a randomly selected subset of 1000 of the total 4220 papers from the ResearchIndex, as running experiments with the full dataset would have taken too long, as we were continuously tweaking learning parameters and instance representations. As before, we ran binary experiments over the different classes, but with Progol we attempted to learn a classification for each of the 17 classes. We used a single Prolog predicate of the form *inClass(+article)* to represent class membership. This became the target clause which Progol would try to learn.

[8] Progol is freely available online from http://www.doc.ic.ac.uk/~shm/Software/.

4.4 RDF as 1st Order Logic

Initially we chose a very simple approach to map RDF to Prolog. As the RDF data-model represents *(subject, predicate, object)* triples, we employed a single Prolog predicate called *triple*. Figure 14 illustrates our initial representation and corresponds to the RDF appearing in Figure 4. Note how the BibTex type maps to a RDF concept within the namespace of these experiments.

```
triple( url, zucker92performance,
    'citeseer.nj.nec.com/zucker92performance.html' ).
triple( booktitle, zucker92performance, 'Proceedings of the 19th
    International Symposium on Computer Architecture' ).
triple( type, zucker92performance,
    'http://www.csd.abdn.ac.uk/~ggrimnes/exp/#inproceedings' ).
triple( address, zucker92performance, 'Gold Coast, Australia' ).
triple( title, zucker92performance, 'A Performance Study of
    Memory Consistency Models' ).
triple( year, zucker92performance, '1992' ).
triple( author, zucker92performance, 'R. Zucker and J.-L. Baer' ).
```

Fig. 14. RDF Encoding – Initial Approach.

Perhaps as expected, this approach did not give very good results as the search space for Progol became extremely large. Due to Progol only having one predicate to use in the construction of the result clause, the algorithm would quickly get lost down a faulty path of the search-tree with incorrect constants or incorrect unifications, and never recover.

Our first improvement was to change the way we represented triples. Instead of casting them all to the same predicate, we created a Prolog predicate corresponding to each RDF predicate, e.g. *triple(author, zucker92performance, 'J. Zucker')* became *author(zucker92performance, 'J. Zucker')*. Secondly, we recognised that Progol operates on strings and if two literals are not *exactly* equal, Progol has no way of generalising over them. This led us to preprocess all strings so that each word became a separate Prolog fact. For example, instead of *title(learning02grimnes, 'Learning from the Semantic Web')* we would have: *title(learning02grimnes, 'learning'), title(learning02grimnes, 'semantic') and title(learning02grimnes, 'web')* . In addition to this simple pre-processing we also applied a list of synonyms for commonly used abbreviations and misspellings, e.g. *proc, procs* and *proceeding* all map to *proceedings*, *sixth* maps to *6th*, etc. Finally, we standardised the representation of author names to first initial plus surname, as BibTex does not specify a standard. This means that *Alun Preece, Preece A.* and *Alun D. Preece* all map to *A. Preece*. As with title words we would created one Prolog fact for each author. An example of our final representation appears in Figure 15.

```
url( zucker92performance, 'citeseer.nj.nec.com/zucker92performance.html' ).
booktitleword( zucker92performance, 'proceedings' ).
booktitleword( zucker92performance, '19th' ).
booktitleword( zucker92performance, 'international' ).
booktitleword( zucker92performance, 'symposium' ).
booktitleword( zucker92performance, 'computer' ).
booktitleword( zucker92performance, 'architecture' ).
type( zucker92performance, 'http://www.csd.abdn.ac.uk/~ggrimnes/exp/#inproceedings' ).
address( zucker92performance, 'Gold Coast, Australia' ).
titleword( zucker92performance, 'performance' ).
titleword( zucker92performance, 'study' ).
titleword( zucker92performance, 'memory' ).
titleword( zucker92performance, 'consistency' ).
titleword( zucker92performance, 'models' ).
year( zucker92performance, '1992' ).
author( zucker92performance, 'R. Zucker' ).
author( zucker92performance, 'J. Baer' ).
```

Fig. 15. RDF Encoding – Second Approach.

4.5 Results

Lack of space prevents us from presenting the Progol results in full. However, we will discuss some features of the results and will provide illustrative examples. For most classes the majority of the rules discovered by Progol are of the form: *inClass(zucker92performance)*, meaning that Progol was unable to find any common features between instances of the given class, and simply returned an *inClass* clause that lists all the instances declared to be in that class. This problem is almost certainly caused by the small number of features used to describe each instance, and overlap between some of the classes, making it difficult for the algorithm to identify discriminatory generalisations.

Despite this problem, some rules were discovered that covered more than a single instance. For example, Figure 16 shows the rules generated from the *Agents* class; the first five rules are straightforward, and encapsulate obvious facts about publications in the area of *agents* technologies. However, the sixth rule, *inClass(A) :- titleword(A,bdi).*, is more interesting. BDI is an abbreviation for *beliefs, desires and intentions*, a common paradigm within agents research [19]. An active researcher in the agents field would find this almost as obvious as the other rules, based on their knowledge and experience of the field. This Progol result is thus a piece of general knowledge, which is not only usable in trying to classify new research papers from the ResearchIndex, but could also potentially be applied outside this experiment. Several of the other classifications also generated rules of a similar type. We find these results from our Progol experiments very exciting, and present a selection in Figure 17. The rules range from the slightly bizarre, such as *all papers published in volume 18 are Theory*, to rules containing interesting and potentially reusable knowledge, such as *Papers published by Morgan Kaufmann appearing in books with learning in the title are in the field of Machine Learning.*

Examination of the Progol results indicate that Progol is overfitting to the problem, by creating a large number of *inClass* rules. This is because it is im-

```
Agents:

inClass(A) :- author(A,'A. Rao').
inClass(A) :- author(A,'D. Lambrinos').
inClass(A) :- titleword(A,agent), titleword(A,mobile).
inClass(A) :- type(A,'http://www.csd.abdn.ac.uk/~ggrimnes/exp/#misc'),
textword(A,agent), titleword(A,agent).
inClass(A) :- year(A,1999), titleword(A,agents).
inClass(A) :- titleword(A,bdi).
```

Fig. 16. Excerpt of Progol Results - Agents Experiment

```
Artificial Intelligence:
inClass(A) :- journal(A,'SIAM Journal on Control and Optimization').
inClass(A) :- journal(A,'Computational Linguistics').

Databases:
inClass(A) :- titleword(A,warehousing).
inClass(A) :- titleword(A,deductive).
inClass(A) :- titleword(A,aggregate).
inClass(A) :- titleword(A,transactions).

Machine Learning:
inClass(A) :- publisher(A,'Morgan Kaufmann'), booktitleword(A,
learning).
inClass(A) :- titleword(A,based), titleword(A,case).

Programming:
inClass(A) :- pages(A,225), booktitleword(A,conference).


Security:
inClass(A) :- booktitleword(A,privacy).
inClass(A) :- titleword(A,watermarking).
inClass(A) :- titleword(A,encryption).

Theory:
inClass(A) :- volume(A,18).
```

Fig. 17. Progol Results - Sample Rules

	ML	**AI**	**IR**	**HCI**	**DB**	**Agents**
Recall:	62.50%	58.93%	26.92%	45.31%	37.50%	58.93%

Fig. 18. Recall for Pruned Progol Rules - ResearchIndex Dataset.

possible for Progol to generalise any further without creating rules that are not correct for 100% of all the instances. It would be desirable to allow rules that would correctly classify, say, 99% of the instances, thus allowing more generalisations and pruning the set of resulting rules. By doing this we would hopefully get a smaller set of rules for each class, and a much smaller set of *inClass* statements. However, as Progol has no built in method for doing this, we chose to take a very simple approach as follows: all the *inClass* rules were discarded and only the more knowledge-rich rules retained for each class. When this reduced set is used for classification, the precision of the rules is still 100% as no articles will ever be incorrectly classified, however recall will no longer be perfect. The percentage of recalled instances for each class are presented in Figure 18.

5 Related Work

We are aware of little work concerned with application of machine learning to Semantic Web data. This is in contrast to applications to the Web, of which there have been many. For example, Syskill & Webert [17] uses machine learning to acquire a model able to predict which links on a Web page a user will find useful. It does this by analysing a set of Web pages manually rated by a user, which are then processed using structural IR techniques. Syskill & Webert uses a Naïve Bayes classifier, but the authors also report investigations using nearest neighbour algorithms, ID3, perceptrons and multi-layer neural networks. Webwatcher [1, 12], like Syskill & Webert, is a browsing aid which attempts to annotate a Web page with information on what links a user might find useful. The authors explored a variety of learning algorithms, such as Winnow [9] and Wordstat. Underlying the system was an instance representation which did attempt to exploit more of the structure of the HTML documents, as link text, headers, etc. were treated differently. Letizia [8] is a browser helper, displayed in a separate window next to the user's browser. It pre-fetches all outgoing links from the current page and will do a breadth first search to advise the user on which links to visit next. Letizia uses TF/IDF to extract content from pages, and uses the weighted terms to identify documents matching the user's interest.

6 Conclusions

Our results have demonstrated that today's available Semantic Web markup cannot be expected to outperform conventional machine learning applied to plain text, with regards to accuracy of the learned model. However, it must be noted that applying the same algorithm to the full text of an article of 6000 words, and to 10 lines of RDF code, while still getting equally good predictive accuracy does constitute an increase in performance and scalability. In the Web context this is especially important as algorithms will be expected to scale to millions of pages. Nevertheless, we remain somewhat unsatisfied with our current results for a number of reasons. Although we attempted to find real Semantic meta-data, we admit that we are not completely happy with our datasets. The ITTalks dataset is too small to be able to draw any firm conclusion from any results, and the ResearchIndex dataset has generated meta-data from a source which was never meant to provide real *meaning.* Also, when the Semantic Web becomes reality, many supporting technologies should be available, most significantly ontological support and the availability of inference engines, which should allow for easy generalisations of the kind: *A*, Simulated Annealing and Depth-first are all types of Search algorithms* which could be used to facilitate classification tasks such as the ones we have attempted in this paper, but which are nearly impossible to discover without any background information.

6.1 Future Work

We plan to continue exploring issues concerned with Progol and Knowledge Intensive Learning on the Semantic Web, primarily attempting to utilise background information to help Progol generalise better over classes. We plan to explore generation of such background information from ontologies referenced in the meta-data, as well as through the use of general background information such as as synonyms or word similarity.[9].

We are very interested in trying to apply the resulting Prolog clauses outside the original experiment. As the results are first order logic it should be possible to map them back to a representation in RDF or a similar logic based format, thus exporting the *model* that was learned.

Due to the current shortcomings of Semantic Web data we plan to do further experiments with our current datasets. Primarily we intend to re-classify the ResearchIndex papers based on personal interest, thus moving further towards the personalisation and learning user models scenarios used as motivation for this work. We would also like to run Progol with the full ResearchIndex dataset, not just a small subset of the instances, as well as attempting to find a good way of mapping DAML+OIL to Prolog, so that we may run Progol experiments with the ITTalks dataset.

References

1. Robert Armstrong, Dayne Freitag, Thorsten Joachims, and Tom Mitchell. Webwatcher: A learning apprentice for the world wide web. In *AAAI Spring Symposium on Information Gathering*, pages 6–12, 1995.
2. Tim Berners-Lee. What the semantic web isn't but can represent. http://www.w3.org/DesignIssues/RDFnot.html, 1998.
3. Scott Cost and Steven Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10:57–78, 1993.
4. S. Dudani. The distance-weighted k-nearest-neighbour rule. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-6(4):325–327, 1975.

[9] This data could for example be taken from WordNet (http://www.cogsci.princeton.edu/~wn/)

5. Paul W. Finn, Stephen Muggleton, David Page, and Ashwin Srinivasan. Pharmacophore discovery using the inductive logic programming system PROGOL. *Machine Learning*, 30(2-3):241–270, 1998.
6. Salton G and Buckley C. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523, 1988.
7. Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of International Conference on Machine Learning*, pages 331–339, 1995.
8. Henry Lieberman. Letizia: An agent that assists web browsing. In Chris S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 924–929. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 20– 25 1995.
9. N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.
10. D. Michie, D. Spiegelhalter, and C. Taylor. *Neural and Statistical Classification*, chapter 70, pages 1297–1300. Ellis Horwood, 1994.
11. T. Mitchell. *Bayesian Learning*, chapter 6, Machine Learning, pages 154–200. McGraw-Hill, 1997.
12. D. Mladenic. Using text learning to help web browsing. In *Proceedings of the 9th International Conference on Human-Computer Interaction.* HCI International 2001, New Orleans, LA, USA, 2001.
13. S. Muggleton. Inverse entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4):245–286, 1995.
14. S. Muggleton, R. King, and M. Sternberg. Protein secondary structure prediction using logic-based machine learning. *Protein Engineering*, 5(7), 1992.
15. Stephen Muggleton and John Firth. Cprogol4.4: A tutorial introduction.
16. Terry Payne. *Dimensionality Reduction And Representation For Nearest Neighbour Learning.* PhD thesis, University of Aberdeen, 1999.
17. Michael J. Pazzani, Jack Muramatsu, and Daniel Billsus. Syskill webert: Identifying interesting web sites. In *Proceedings of the American National Conference on Artificial Intelligence, Vol. 1*, pages 54–61, 1996.
18. M Porter. An algorithm for suffix stripping. Technical Report 14, Program, 1980.
19. A. Sloman and B. Logan. Architectures and tools for human-like agents, 1998.
20. A. Srinivasan, R. D. King, S. Muggleton, and M. J. E. Sternberg. Carcinogenesis predictions using ILP. In S. Džeroski and N. Lavrač, editors, *Proceedings of the 7th International Workshop on Inductive Logic Programming*, volume 1297, pages 273–287. Springer-Verlag, 1997.
21. M. Stone. Asymptotics for and against cross-validation. *Biometrika*, 64:29–35, 1977.

Semantic Methods and Tools for Information Portals
The SemIPort Project

Jorge Gonzalez-Olalla, Gerd Stumme

Institute of Applied Informatics and Formal Description Methods AIFB,
University of Karlsruhe, D–76128 Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de/WBS
{jgo, stumme}@aifb.uni-karlsruhe.de

1 Objectives and Research topics.

The goal of the SemIPort project is to develop innovative methods for presenting, storing and accessing scientific information within a Semantic Information Portal. The project will enable users to access a semantically structured information inventory, while allowing them to integrate their own information.

Several scenarios as the AIFB portal and the DBLP online bibliography will be used to test the developed methods. Later, the results of the project will be implemented in the form of an Internet Portal for the German Informatics Society (GI) as a competency and service network .

The project is financed by the German bmb+f for a period of 3 years starting June 2002. The participants are the Institute for Applied Informatics and Formal Description Methods at the University of Karlsruhe (coordinator), the German Research Center for Artificial Intelligence, the Fraunhofer Institute for Integrated Publication and Information Systems, and the Data Bases and Information Systems Group at the University of Trier.

The **research tasks** within the project are:

(1) Ontology Modelling and Metadata Standards. (2) Scalable Storing, Processing and Querying of Integrated XML and RDF Inventories (Knowledge Warehouse). (3) Web Mining and Ontology-based Knowledge Integration. (4) Visualization and Browsing of Complex Data Inventories. (5) Personalization and Agent-based Interaction.

The most related to the workshop is task (3). This topic will concentrate on:

i) Semantic Web Content and Structure Mining. The explicit semantics of semantic web data will be used to improve the results of ’classical’ web and Data Mining techniques. E.g. for ’hubs and authorities’ one could distinguish between different kinds of authorities.

ii) Semantic Web Usage Mining. The behavior of the portal’s visitors will be used to optimize the portal’s structure and usability, by developing tools that will help to manage and update the ontology according to the actual needs of the users. A ’semantic log file’ will be created and analyzed for this purpose.

iii) Flexible techniques for knowledge integration will be developed. Existing methods of schema integration for relational data bases and ontologies will be analyzed and expanded for dynamic integration.

Basic Techniques for the Extraction and Annotation of Machine Understandable Information

Manuela Kunze, Dietmar Rösner
Otto-von-Guericke-Universiät Magdeburg
Institut für Wissens- und Sprachverarbeitung
P.O.box 4120,
39106 Magdeburg, Germany
{makunze, roesner}@iws.cs.uni-magdeburg.de

June 28, 2002

Project Description

The core of the semantic web are machine understandable information about resources (mostly these are documents). Currently the documents available in the www are mostly without any explicit information about their content. In our project we want to support the automatic extraction of information and annotation of documents with machine understandable information. Our document suite XDOC provides a collection of tools for the flexible and robust processing of documents in German. The analyses of documents are based on linguistic methods and techniques of knowledge extraction. Presently we deal with the document content for the following application fields:

- automatic annotation of documents with metadata e.g. in a kind of RDF(S) and
- extraction of company profiles from webpages[1].

The document suite contains tools for preprocessing, structure and POS tagging, syntactical parsing, and semantic analysis[2]. The latter includes methods like the semantic tagging of unique words (tokens), a case frame analysis and the mapping from syntactic structures into semantic relations. As resources several lexica for linguistic and semantic analysis are used. Most of these lexica are manually recorded but we also implemented techniques for automatic acquisition of entries for the semantical lexicon resp. for possible relations inside an ontology[3]. The XDOC collection of tools is based on the use of XML as unifying formalism for encoding input and output data, resources (e.g. lexica) as well as process information. Through XSL transformations different views of the results are generated, in particular for the annotation with metadata. We can present these information as a Topic Map or in RDF(S) or in another KR language. Through the use of XML as internal format we are independent of the finally required target markup language.

References

[1] S. Krötzsch and D. Rösner, "Towards extraction of company profiles from webpages," in *2nd International Workshop on Databases, Documents, and Information Fusion*, (Karlsruhe, Germany), July 2002. to appear.

[2] D. Rösner and M. Kunze, "Die Document Suite - XML-basierte Sprachverarbeitung als Basistechnologien für das Semantic Web," in *XML Technologien für das Semantic Web - XSW 2002* (R. Tolksdorf and R. Eckstein(Hrsg.), eds.), no. P-14 in Lecture Notes in Informatics (LNI) - Proceedings, (Berlin, Germany), pp. 119–133, GI-Edition, Köllen Druck + Verlag GmbH, Bonn, June 2002.

[3] D. Rösner and M. Kunze, "Exploiting sublanguage and domain characteristics in a bootstrapping approach to lexicon and ontology creation," in *Proceedings of the OntoLex 2002 - Ontologies and Lexical Knowledge Bases at the LREC 2002*, (Las Palmas, Canary Islands), May 2002.

Semantic Web and Retrieval of Scientific Data Semantics

Goran Soldar
School of Computing and Mathematical Sciences
University of Brighton
g.soldar@brighton.ac.uk,

Dan Smith
School of Information Systems
University of East Anglia
djs@sys.uea.ac.uk

Project Description

Introduction

The results of scientific activities that include observations, experiments, interactions, deductions, etc. are stored in data sets and often made available to the scientific community. These large volumes of scientific data are typically kept and managed in an ad-hoc manner and it requires a substantial effort to discover and evaluate data quality and suitability for a particular analysis. The provision of appropriate metadata and links to related resources enables the possibility of intelligent assistance in finding and evaluating data sets for scientific research. Such tools are of increasing importance as the volume of scientific data is increasing rapidly due to the extensive deployment of automated data collection and monitoring systems. To find and process scientific data sets available on the Web is time consuming despite the fast and powerful search engines. For example to find files related to temperatures we used the Google search engine. The key word "Temperature" was entered and search restricted to the University of East Anglia Climate Research Unit (`http://www.cru.uea.ac.uk`). Google returned 773 pages. For a scientist browsing every single web page to find required data sets, without guarantee that they will find what they need, is simply too much time consuming exercise. It would be desirable that data files are described in a way that they could be quickly found and their semantics learnt automatically by a machine. The framework for such a semantic retrieval of scientific data is offered by W3C in the form of Resource Description Framework (RDF) [1,2].

This project seeks to address methods and provide an architecture for describing, managing, retrieving, and extracting semantic information from specific science domains. The conceptualization of the subject domains including the development of the vocabulary [3], is performed using RDF model and schema constructs. We distinguish between two levels of metadata related to RDF, *Instance Metadata* and *Schema Metadata*. The description of a particular resource using RDF constructs is known as the instance metadata, whereas Schema Metadata describes a particular ontology The notion of a semantic case is being introduced to capture semantic forms. The heterogeneity of data sources is exposed and integration of data is achieved through a mediation process [4]. The primary goal of our project is to enable machine processing of semantic information that would be utilized to query the actual information from data sets To address the problem of extracting semantic information from data files, we build an ontology for the meteorology domain which is then use to create semantic cases as file description templates. We use RDF Model Syntax and RDF Schema to create semantic cases instances. The architectural infrastructure is based on the Semantic Retrieval Model [5].

Storage and Management of Ontologies

The management of RDF structures is an open issue, and W3C does not recommend any particular method for manipulating such data. By definition RDF Model is set of triples. This property can be utilised to achieve manipulation of RDF triples as a Relational Model. The storage and manipulation of RDF graph structures represents a problem. It does not matter whether the data is kept in RDF syntax or as triples, the problem is that there is no RDF data management system that provides access and manipulation of ontologies. In addition the existing RDF parsers are design to work on single documents only, so the problem is how to access documents remotely, and how to achieve the maintenance either of parts or of the whole documents in terms of modification, insertion, and deletion. Another question that arises is whether there is a need to keep whole RDF documents at all. Since modern database management systems are based on the concept of the Relational Model it is considered to use the existing RDBMS for manipulation of RDF data or alternatively to build RDF native data storage model.

Retrieval and Extraction of Semantics (RDF Triple Engine)

The system is based on client-server architecture that includes specialized RDF servers. The advantage of this approach is in data management facilities of RDBMS that are utilized for the manipulation of raw RDF data. The architecture comprises the 4 layers: 1) *Interface Layer*, 2) *Web Infrastructure Layer*, 3) *RDF Management Layer* and 4) *DBMS Layer* The interface layer provides access to semantics descriptions to human users as well as to application programs. The prototype of the Semantics Retrieval Language (SRL) is currently being developed to provide semantics-orientated retrieval of DBMS managed RDF data. QBE-like Web form is used to assist users in specifying their requests although SQL-like statements are also supported. The interface handlers (access rights and authentication) are built as Java servlet modules and they run inside the Apache Tomcat servlet engine, which is seamlessly connected to the Apache Web server. All valid requests are transparently forwarded to the Semantics Support Server (SSS). Since SSS runs as a separate service, applications can establish direct connection with this server using the internal SRL protocol over TCP/IP connection.

RDF Triples Engine (RTE) is a module responsible for manipulating triples and executing semantic queries. The check for namespace existence for each prefix is performed by RTE before the records are inserted into the database. A user with no knowledge of graph structures and its specific elements (containers) will find it difficult to assemble the full picture from the output. This problem is solved by adding the additional query processing to RTE, which is aware of RDF semantic and also is able to produce results suitable for human use (HTML document that conforms to XHTML structure) or for further processing (XML structure).

References

[1] WWW Consortium "Resource Description Framework (RDF), Model and Syntax Specification", Available at: *http://w3.org./TR/REC-rdf-syntax.*

[2] WWW Consortium "Resource Description Framework (RDF) Schema Specification", Candidate Recommendation, Available at: *http://w3.org./TR/1999/REC-rdf-schema.*

[3] T. Gruber "Towards Principles for the Design of Ontologies Used for Knowledge Sharing", *International Workshop on Formal Ontology, Padova, Italy*, 1993.

[4] D. Smith, M. Lopez, "Information finding and filtering for collections of semi-structured documents", *Proc. INFORSID XV*, Toulouse, 353-367, 1997.

[5] G. Soldar, D. Smith "Retrieving Semantics from Scientific Data: A Domain Specific Approach Using RDF", *Proceedings of the IASTED International Symposia on Applied Informatics*, Innsbruck, Austria, February 2001

Data Fusion and Semantic Web Mining: Meta-Models of Distributed Data and Decision Fusion

Vladimir Gorodetski, Oleg Karsaev, Vladimir Samoilov
St. Petersburg Institute for Informatics and Automation
{gor, ok, samovl}@mail.iias.spb.su

Abstract:

According to the Project funded by European Office of Aerospace Research and Development (AFRL/IF) we are developing mathematical model, multi-agent architecture and technology realized as a software tool supporting design and implementation of Data Fusion (DF) applications of broad spectrum. A multitude of tasks to be solved with regard to the development of DF software tool can practically be divided into two groups. The tasks whose solutions make use of methods, models and technologies of other adjoining scientific fields, for instance, data mining and knowledge discovery, multi-agent systems, object-oriented design, etc fall into the first group. The second group includes the tasks specific for DF systems and require development of specific methods, models and technologies. In fact, the most part of tasks of the last group fall into the interests of Semantic Web Mining. Although the tasks of both above groups are the subjects of the Project, below the DF specific tasks are only highlighted.

In its essence, DF task is one of making decisions (as a rule, classifications) on the basis of distributed data sources presented by distributed databases with access through Intra- or Internet. These sources contain data that can be represented by different data structures (temporal, sequential, transactional, relational), they can be of different physical nature (images, signals, truth values, etc) and measured in different scales (Boolean, categorical, real), be of different accuracy and reliability, they can be uncertain, contain missing values, etc. The objective of a DF system is to combine useful information from all of these sources to make decision, for instance, classification of an object, object state, situation, etc.

Within DF specific tasks two classes of them are of most significance. The first is development of *meta-model of distributed data* sources and the second is development of *meta-model of combining decisions* produced on the basis of particular sources. Note that the former task is purely Semantic Web related.

Three main issues of DF specific R&D of the Project are the subjects of presentation.

1. The *ontology-centric approach* which is considered as a basis of the development of meta-model of distributed data sources. In particular, ontology-based approach aims to answer the following questions associated with the development of meta-model of distributed data sources:

How to resolve the data non-congruency problem caused by *heterogeneity* and *distribution* of data sources? The particular questions are: How to provide for monosematic understanding of the terminology used in formal specification of distributed entities which, as a rule, are developed by distributed analysts? How to solve the entity identification problem, which arises due to the fact that the same entity specification is represented by its fragments in distributed databases? How to cope with the diversity of data physical natures, scales of attribute measurement, variety of data accuracy, duplication of the same attributes in different data sources? How to provide compatibility of ontology-based specifications of DF system notions and their interpretations represented in terms of a database language?

2. The second important issue of DF system design is *distributed learning* and meta-model of combining decisions. The questions to be answered here are as follows: What structures are used to combine particular decisions made on the basis of particular data sources to generate global decision? How to manage distributed data in order to provide correctness with regard to allocation of training and testing data used for learning particular classifiers and how to form and manage meta-data used for learning components responsible for combining decisions? What formal techniques are appropriate to combine decisions in accordance with the hierarchy of classifiers?

3. The third issue is of architectural kind. We use multi-agent architecture which to our opinion is the most appropriate for the implementation of DF systems and also for many other Semantic Web-based applications. This architecture includes specific components (intelligent agents) and protocols aiming at solving the questions formulated within both above issues.

In presentation we intend to highlight our results concerning to the above issues and their correlation with Semantic Web Mining-related tasks and problems. In addition, we intend to present an outline of a technology supported by a software tool used for design and implementation of DF software tool including design and implementation of its ontology component. In conclusion a brief outline of DF applications developed and being developed can be outlined.