Ontology Extraction for Educational Knowledge Bases*

Peter Cassin, Chris Eliot, Victor Lesser, Kyle Rawlins, and Beverly Woolf

140 Governor's Drive, Department of Computer Science
University of Massachusetts, Amherst, MA 01003-4610, USA
{pcassin, eliot, lesser, rawlins, bev}@cs.umass.edu

1 Introduction

A student who wishes to learn about some particular topic does not have many options. An often used tool is the search engine, which gives a tiny and difficult to control window into the vast amounts of information that is available on the Internet. A student who wants to learn some concept should be able to interact with the available information in a coherent and personalized way. The classroom is the ideal of this goal, and our system would not replace, but augment it. It is within the reach of modern tutoring systems to use both knowledge of the student and of the subject's structure in order to present a subject in a manner that is more coherent and pedagogically sound than currently existing technology. One of the basic building blocks of such a system is the model of topic structure, and most importantly, how to obtain the information that fills the model.

Here we outline our research platform for the study of ontology life-cycle management, as well as several techniques that have so far had qualitative success. This research is taking place within the context of the Digital Libraries Initiative, under which thousands of instructional objects are organized, ranging from multimedia tutors [1], to lecture notes and papers. Our long term goal is to develop agent based tutoring systems which draw on this large knowledge base, and we have discussed our approach to this in other recent work [2, 3, 4].

There are two main components to this paper. First, we describe our architecture for extracting structured information from raw web pages. Second, we describe our techniques for extracting a more complete ontology of pedagogical information from the structured information.

Available online course materials range from short syllabi, to detailed breakdowns of the course with syllabi, lecture notes, and sometimes even textbooks online. Often, this information exists in disparate formats, and often contain

* This material is based in part upon work supported by the National Science Foundation under Grant No. IIS-9812755. This work was supported in part by the Center for Intelligent Information Retrieval and in part by the National Science Foundation Cooperative Agreement number ATM-9732665 through a subcontract from the University Corporation for Atmospheric Research (UCAR). Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

L. van Elst, V. Dignum, and A. Abecker (Eds.): AMKM 2003, LNAI 2926, pp. 297–309, 2003.

large segments of free text or prose. In order for agents to draw effectively on this collection of information, we would like to find knowledge structure (ontology) that exists at a deeper level within the collection. This is a synthesis of data mining, extraction and fusion. It is embedded this second perspective that we are approaching the research, and this lends itself well to a multi-agent perspective.

Our knowledge structure is implemented in terms of an ontology represented in a database. We take this to be a structure consisting of *topics* or (interchangeably) *concepts* and the *relationships* between these topics.

For our purposes, the internal structure of a topic differs from the common formulations in existing computational work on ontologies [5] and ontology extraction [6, 7, 8]. In these applications the goal is often much more lexically oriented, or more domain driven. In our system, the goal is domain driven but fairly abstract in that we want to represent ontological relationships that deal with learning. It would be more correct to refer to our notion of a concept as a *pedagogical concept*.

The data sources that we use are mostly course materials found online, and consist of things like course syllabi, textbook chapters, lecture notes, etc. These are found in unstructured form (as simple web pages, usually, or text), and our goal is to discover structure from their text. We have done a certain amount of theoretical research as to how this might be done, as well as some implementation of basic ideas.

Basic Assumptions One of our most important assumptions is that there is a conventional or normative pedagogical structure that is assumed about certain materials, rather than there being any absolutely true structure. Instead of telling educators what to do, we are attempting to describe what they do, and how they implicitly relate topics.

For example, we assume that when different instructors write a syllabus or textbook for a Data Structures class, there are patterns that are independent of the instructor. These patterns follow from an consensus as to how the topic should be taught. This consensus is not the 'best' way to teach something, but rather a convention. Embedded in such conventions is deeper structure of how knowledge is related.

2 Preliminary Data Extraction

Prior to the ontology maintenance cycle, significant work is necessary to transform unstructured data into the structured form that is taken into the ontology. Since the data sources are mainly raw web pages, we have put much effort into various Information Retrieval techniques to find structure in these.

The goal of this part of the process is to find ontologically meaningful topics in existing pedagogical materials on the World Wide Web. The problem is similar to the Information Retrieval problem of topic detection.

2.1 Discovery of Topics

Many information sources can be analyzed to extract ontology topics on the web. At many educational institutions students depend upon course web pages as a critical resource for the educational process. The distribution of course subjects which use web pages is currently much less skewed toward computer-related courses than it was previously, and these pages provide a broad and relatively deep information source.

Course web pages often provide a rich source of varying kinds of information about courses. Almost all information is useful to our goal, but some kinds are more useful toward immediate progress than others.

Some sites offer minimal information, such as specifics about some course's meeting times, location, and grading scheme. This information is very useful to students in the course, and is potentially useful in studying how different courses fit together at a university. However, for immediate needs of topic detection, this is not useful to us.

Many course web pages, however, now contain online versions of lecture notes, syllabuses, homework assignments, and other rich sources of data. These pages (and the media they contain) can provide a sound basis for extraction topic information. In addition, the semi-structured nature of HTML pages (i.e. the formatting tags used) provide additional potentially useful cues for extracting information; cues which are not present in plain-text sources.

Extracting this information is, unfortunately, not an easy task. A given course web page typically has high intra-page consistency, meaning that the information is organized in a regular fashion within one author's collection of pages. For example, if an author wishes to enumerate textbooks for a course, they will most likely use one particular formatting structure for that entire enumeration (be it a table, ordered list, unordered list, etc.). They are not likely to use a list bullet to mark the first textbook, a table cell for the second, and a numbered element of a new list for the third. In comparison, there is very little inter-page formatting consistency, on the lowest level, which can be derived across different pages. Since HTML is a text based markup language, there are many ways to construct pages. Even the general means vary; an author could use one of many publicly available HTML editors (commercial or free), automatically generate them using third-party applications from another data source (i.e. a more structured format such as an XML document designed for marking up course information), or write the pages by hand in a plain text editor.

In addition, extracting information from web page structures such as tables is not a negligible issue by any means. Rather, it poses a problem in its own right, and has been the topic of considerable research [9, 10]. For example, table orientation is never a given - though to a human it is immediately evident whether the information contained in some table is row or column oriented. This fact is very difficult for an algorithm to determine, because it needs knowledge about the underlying information in order to determine. In addition, tables may be used to obtain format effects that are not semantically significant.

2.2 Extraction Architecture

We have developed an architecture for extracting the ontological information that is necessary, and in particular for studying topic detection in course web pages. A description of some of the important general characteristics of the architecture follows.

Though there is implied sequence in the different steps of this process, they are not intrinsically tied to each other. Rather, the system runs in a soft-serial order. Each process, while dependent on the existence of a particular kind of information, runs independently. As long as something feeds it that information, or it has a backlog of information to process, it will continue. The architecture is designed for continuous operation, where different components will be scheduled to run at specified times, and once complete, the topic-detection process will happen in an automated fashion.

The architecture is dependent on the idea of iterative improvement. Because all components can be retrained on new and better data, we are working on an interactive process by which the user can inspect the current results of any of the steps in the process, and use correct results for retraining. While the system will run without this step, it is important especially for initial development. The overall structure of the system, including both halves of the process, is describe in figure 1.

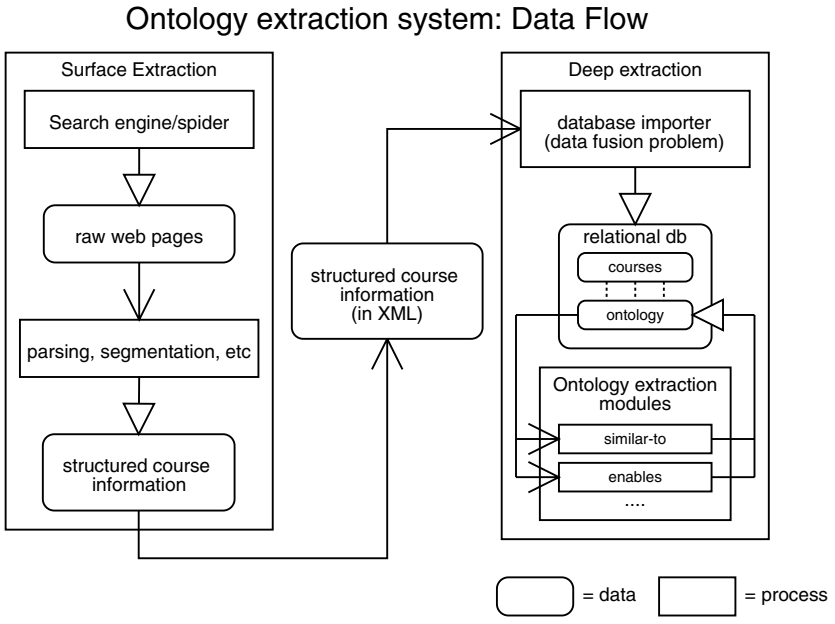

Fig. 1. Broad structure of the architecture

Web Spider The first step in the process is simply to find the Internet-based educational resources. The web spider does this, by downloading and storing pages locally. We configure the spider to automatically download a fixed number of pages per day, to serve as input for the later stages. This is completely implemented, and needs nothing more than some improvement in search heuristics.

Classifier As mentioned earlier, there are many different kinds of web pages with useful information. Although it might be possible to extract from all pages uniformly, we have decided to apply a document classifier first, to simplify the extraction process. Despite wide variance in the organization and layout of content in web pages with extractable information, they typically fall into one of several categories. These categories include:

Particular course pages These provide information on the course, as well as textbooks, meeting times, etc. Pages such as syllabus pages fall into this category.

Department course pages These typically provide short descriptions for every course in a department, school, or even university.

Unrelated pages These pages provide little to no cues as to educational content.

By attempting to distinguish ahead of time what "type" of page we are looking at, we make the job of extraction easier in several ways. Most importantly, the classifier filters out most irrelevant pages quickly. These pages make up most of the web, and something as computationally expensive as the extraction algorithm should not be wasted in the first place on these. In addition, we can then use the classification of a page to bias the extractor toward expecting certain kinds of information. In the future, we are considering more fine-grained classifications, such as separating out the different kinds of individual pages that one course might use.

While it may have been possible to incorporate this processing directly into the extractor, by externalizing the process we allow greater modularity, and the potential for parts of the system to continue if others fail.

Our classifier is implemented using the MALLET system, developed mainly by Andrew McCallum and implementing a variety of text classification and extraction techniques [11, 12, 13].

The implementation of the classifier is at a usable stage; our current task is improving the classification heuristics and providing a broader cross-section of training data.

Extractor The key step in converting raw semistructured web data into the structured ontological form is the step of correctly extracting pedagogical topics and other important information into a more structured form. This is an extremely difficult IR problem, and here we are again using the MALLET system rather than trying to invent our own solution.

The extractor builds a wrapper around a raw page, aiming to locate and tag useful information for the ontology extraction process. The types of useful information available vary by the nature of the page inspected. For course web pages, the following pieces of information are the ones sought:

- Course name
- Course number
- University
- Prerequisites
- Textbooks
- Professor
- Syllabus
  - Ordered topics on some level of granularity (lecture, unit, etc)
  - Homework assignments
  - Lecture notes
  - Readings
- Course description

By far the most challenging piece of information to identify and correctly extract is a course syllabus. This involves determining not only if and where it occurs in the data, but segmenting the information internal to the syllabus. Most of the other "interesting" fields have little if any internal data. This is also contains the most important and useful information toward extracting a structured ontology.

For department-wide course description pages, the problem changes slightly. Some of the information that is typically present includes:

- Course name
- Course number
- Prerequisites
- Course description

These pages also involve segmenting different courses from each other. There is also the problem of correlating pieces of information between different pages; do two course numbers refer to the same course? This problem is left for a later stage, where the ontology extraction tries to determine if two pieces of information refer to the same thing.

An earlier version of this system used a completely heuristic extraction process, which proved ineffective. We are working to use Conditional Random Fields[12] as the theoretical basis for the attempt to learn to extract, as it has been used in a variety of domains and is under active testing and development.

The extractor is ongoing work, as it is not a simple problem. However, with the use of the MALLET system, we believe that the solution can be feasibly solved in the short term with useful results.

2.3 Conclusions

We are working toward integration of all the components. Though all components exist in some form, they do not yet usefully communicate. This is true especially of the topic extraction and page classification modules.

We hope to introduce new IR technologies that will help deal with numerous issues:

- Scoped Learning [11]: although there are few formatting regularities which can be used across all web pages (even on a certain topic), there are many regularities which lie within a subgroup of these pages. For example, some universities provide professors with web page templates to ease creation of their pages. Such similarities can be exploited by recognizing the existence of features salient not on a global level, but on some smaller local level. If the scope can be recognized and extraction is successful on one particular page (due to non-formatting relating aspects such as keywords), some of the formatting regularities from that page can be used as cues on other similar pages within the scope. In a sense, our pre-classification attempts to take a similar approach (exploiting noticeable similarities between pages), but a heuristically-based scoped learning approach is much more powerful when applied at a more general level.
- Multi-page structure: Currently, our system does not exploit situations where a course's web page is broken up across multiple HTML files. There is useful hierarchical information here, which would not be extremely difficult to use, regarding the relationship of pieces of information.

With the finished implementation of the technologies described here, we will have a system for taking information from raw webpages and into a useful structured form. While the accuracy will not necessarily be high at this task, preliminary tests show that it will be high enough that, with a large page volume, we will be able to extract enough useful information to feed the ontology extraction process.

3 Overview of the Ontology Extraction Process

On the level of a single agent performing maintenance on the ontology, there are three broad necessary behaviors. The first two are discovery of topics and discovery of relationships. These discovery tasks operate both on new and existing ontology. The third task is the process of analyzing the existing knowledge structure in terms of newly discovered topics and relationships. The existing structure is taken as a hypothesis that is to be tested against newly discovered information. These tasks are independent, and each generates new information that is useful to the other two.

Evaluation An ontology for learning is subjective, no matter how it is constructed. It makes claims about how courses are taught and how information is learned, and we know of no gold standard that is any less subjective than an automatically generate ontology. Still, we are working on a small hand-generated ontology to attempt some evaluation of this sort.

Additionally, we are checking for internal consistency of the process by use of held-out data. That is, the extraction process is run many times from scratch with some randomly selected courses are held aside, and results of these successive runs are compared. This reflects on how sensitive our processes are to the input data, as well as how general the knowledge they are extracting is.

3.1 Topics

The topic is the basic unit of our ontology, and is better thought of as a *pedagogical topic*. At this stage, we are not able to give a theoretical definition of this notion, since that would be a difficult philosophical task. However, we can give intuitive and functional definitions which will suffice for practical purposes.

A topic is some unit of teaching. The granularity or level of abstractness can be quite varied, and can range from the topic of "computer science", down to the granularity of "linked list" or "pointer". These both are parts (meronyms) of "computer science" and of many intermediate topics. Topics are also either typically general categories, or instances of these categories meant for a specific course.

Our working definition of a topic is something that a person might attempt to teach or to learn. We will not pretend to offer a philosophically satisfactory criterion for knowing about a topic, but we feel comfortable judging in specific cases that a student does not know, or needs to know, or wants to know a topic. A teacher or tutor cannot be entirely confident without some assessment mechanism that any instruction has successfully caused learning, but we can at least name specific topics that are the intended goal of a lecture or tutor.

A general formula for education relates prerequisites of concepts, study, and learning. If the prerequisites are satisfied, and study is truly completed, then learning of a topic is achieved. While terms in this formula are somewhat immeasurable, the structure can be taken as axiomatic in principle. Failure to learn can be blamed on failure of study, or unmet prerequisites. The relationships implied by this formula can be taken as constraints on the definition of a topic.

Derived Topics Topics in our ontology can be taken from a number of sources. They could be directly entered by a human, extracted from a course syllabus (or other material), or derived from within the ontology itself. It is this last class that is the most important.

Topics taken directly from a syllabus or table of contents are directly inserted into the ontology, but alone don't provide much information. They are almost uniformly what we consider to be non-abstract topics: specific instantiations of some unknown abstract concept class or category. The more interesting topics

will be the ones that are derived from these instances, which are more abstract, general, and represent a teaching concept rather than some use of that concept.

For example, the topic "linked lists" as found in a syllabus is taken as an instance of the general subject of linked lists. Instantiations of a category may have slightly different details and overlap, but are all referring to the same thing, their category.

We are using a derived topic that approximates a true category, a topic aggregate. When several courses on roughly the same subject material are added to the database, along with these come many topic instances that are closely related, having some kind of similarity relationship. For instance, many Data Structures courses will have a node for "linked lists" in their syllabus, though the position, lecture contents, and wording of that node might vary. A topic aggregate is meant to represent the abstract version of these topics. Every topic in an aggregate is treated as being similar to every other topic in the aggregate. This can be thought of as clusters of similar topics, and the most direct techniques for finding such aggregates are in fact information retrieval based clustering algorithms.

3.2 Relationships

Just as important (or more) than the actual concepts in the ontology are the relationships between these concepts. There are many "classic" ontology relationships that apply to our domain, as well as some that are more specific.

Relationships have certain general properties, many of which can be thought of in algebraic terms, since they are essentially relations on pairs of topics. A relationship could be transitivity, reflexivity, symmetricity, etc. For derived relationships, it is also useful to keep some notion of 'strength' of the relationship, since most techniques we use at this point are probabilistic and do not produce results best represented with a relationship being on or off. Strength in this sense is difficult to define uniformly and the meaning of this value varies depending on the relationship and how it was derived.

At this point in time we have concentrated mostly on extracting just two relationships; one more general, and one that is domain-specific.

Equivalence and Similarity The first relationship we approached started as an *equivalent-to* type of relationship (synonym). Two concepts would be synonyms if and only if they could be freely exchanged for each other in some (or all) contexts. This sort of relationship has some nice properties including being completely transitive and symmetric.

We found, however, that equivalence was not as useful as we expected. At this level of abstraction, true synonyms are comparatively rare but things which seem "similar" to each other are quite common, and we really want to also describe this second class. For example, far more common than finding two instances of a concept that is simply "arrays" in a given course is finding concepts that are more like "arrays of arrays" or "Multidimensional arrays".

In response to this, we started using the notion of a *similar-to* relationship. This for the most part works like the equivalence relationship (with similar properties of opacity), except that it is much less theoretically clean. It is not accurate to say that *similar-to* is transitive or even necessarily symmetric. While two particular topics might be similar, a large ontology might have long chains of similarity. If this were transitive very unlikely things would be considered "similar".

To combat this issue we have introduced a notion of partial transitivity, where the transitivity falls off a certain amount across a relationship as a factor of the strength of the relationship. It is also intuitively desirable to give some weight to a similarity relationship, so to say that some topics are more or less similar. This can be thought of as something like percentage of similarity, but is really an approximation of some (indiscoverable) ideal probability.

Extraction of Similar-to Currently we are using a standard hierarchical clustering algorithm that operates on just the text of a concept. This works reasonably well for small data sets, but may have some problems scaling. The clustering occurs after some processing, including stemming and normalization.

There are some problems with this despite its simplicity. With a small data set it is fairly easy to terminate the clustering algorithm and pick the correct number of clusters, since they simply stop merging at a certain point, so the use of our clustering algorithm requires more study. Also, the clustering is occurring based on word content rather than any notion of semantic content. For our purposes it seems the second is much more ideal, but is unfortunately much harder. One line of approach would be to use WordNet or something like it as a base for semantic analysis.

Concept Prerequisites When learning a particular topic, there are likely to be concepts that would be helpful or even necessary to understand. These can be thought of as concept prerequisites. This is directly valuable information because it gives clues about how one would generate a 'learning sequence' between two arbitrary topics. A related relationship is temporal orderings between two topics in a learning sequence. Such an ordering does not always signal a real concept prerequisite, but can be used as an indication of one. Evidence of what these orderings are is easily found, since most course materials have a somewhat linear organization.

There is clearly some notion of transitivity that is applicable here, though the specifics are not so clear. If A must be learned before B, and B must be learned before C, then it is clear that A must be learned before C. However, it is not necessarily the case that A should be considered a concept prerequisite for C; the issue is one of opacity.

Extraction of Temporal Orderings There are two techniques that we have looked at for inferring this information. The first consists of looking and seeing

whether, on average, instances of two abstract topics are before or after each other in courses where the instances are both found. If they have some ordering which is consistent across many courses, that ordering is added to the ontology as a temporal ordering requirement. If they are not consistent (in some one is after, in some it is before), the information is valuable (they have some kind of independence) but is not currently dealt with.

Our second approach treats the course syllabi as emissions of some Markov model with unknown transition probabilities, and attempts to learn these probabilities. The important orderings in the resulting Markov model are the transitions with high chance that are learned during training.

The results of the first approach to a small selection of Data Structures and Introductory Computer Science courses is shown in figure 2. No post-processing was done on this, except to turn the summaries into English as opposed to stemmed English.

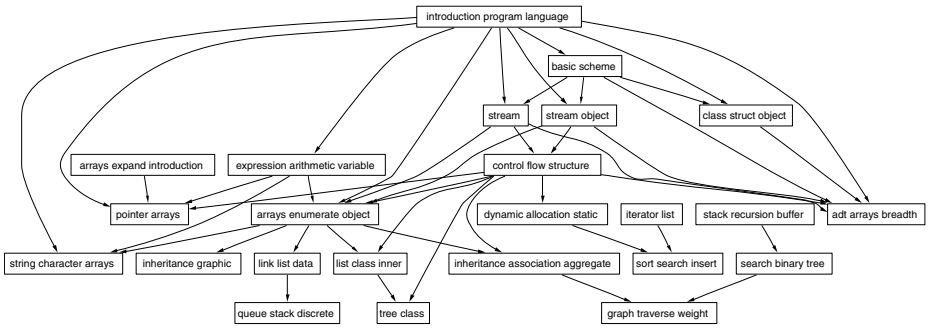

Fig. 2. Extracted Temporal Orderings in Intro CS Courses

Each node in this graph represents what the system has identified as a pedagogical topic. No attempt to differentiate different levels of granularity is made here. The node labels summarize the contents of the topic.

The directional edges each represent an extracted temporal ordering between two topics. They were in fact the highest ranking edges after one iteration of the algorithm, so thousands of other possibilities are excluded in this graph. After several iterations the state may change.

Obviously, neither the topic identification or temporal ordering is perfect. There are several possible reasons for this. The most important is lack of data; we have only a small corpus of training data to work with. The automated topic extraction from web pages is aimed to solve this problem, but many of the odd relationships expressed here are simply relationships that are present in the training data but do not generalize. The second is flaws in the algorithms. One major problem in this category is that topics at different levels of extraction are compared as if they were on the same level. We are working on solutions to this. Finally, this static representation, while making the data useful to humans, hides

much of the underlying complexity of what has been extracted. We are building a dynamic ontology browser to help with this issue.

The results shown in this graph are purely qualitative, not quantitative. A hand-generated ontology would be useful for generating quantitative analysis of the algorithms' results, and development of this is currently in progress. These results are still early, and while our immediate goal is system integration, the next research step is to show quantitative results.

4 Conclusion and Future Directions

We have given a broad overview of our ideas and some of the technology that we have implemented, but many of the techniques are still only in their infancy. In particular, the later stages of the management cycle, where the results of the data fusion are evaluated with respect to the existing ontology are still quite undeveloped.

The next major step is to complete and analyze experimental results for our techniques we are developing, and this is our current task. Already the results are qualitatively promising, and we are close to quantitative evaluations of the ontology management.

References

[1] Woolf, B., Hall, W.: Multimedia pedagogues: Interactive multimedia systems for teaching and learning. IEEE Computer **28(5)** (1995) 74–80
[2] Eliot, C., Woolf, B., Lesser, V.: Knowledge extraction for educational planning. In: Workshop on Multi-Agent Architectures Supporting Distributed Learning, at the 10th International Conference on Artificial Intelligence in Education, San Antonio, TX (2001)
[3] Woolf, B., Eliot, C., Klein, M.: A digital marketplace for education. In Milutinovic, V., Patricelli, F., eds.: Electronic Business and Education: Recent advances in Internet Infrastructures. Kluwer Academic, Norwell, MA (2001)
[4] Woolf, B., Lesser, V., Eliot, C., Eyeler-Walker, Z., Klein, M.: A digital marketplace for education. In: International Conference on Advances in Infrastructure for Electronic Business and Education, L'Aquila, Italy (2000)
[5] Fellbaum, C., ed.: WordNet: an electronic lexical database. Bradford Books (1998)
[6] Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the fourteenth international conference on computational linguistics. (1992)
[7] Caraballo, S.A.: Automatic construction of a hypernym-labeled noun hierarchy from text. Ph.d. diss, Brown University (2001)
[8] Goldman, C., Langer, A., Roschenschein, J.S.: Musag: An agent that learns what you mean. Journal of Applied AI, a special issue on Practical Applications of Intelligent Agnets and Multiagent Technology **11** (1997) 413–435
[9] Freitag, D.: Multistrategy learning for information extraction. In: Proceedings of the 15th International Conference on Machine Learning. (1998)
[10] Cohen, W., Hurst, M., Jensen, L.: A flexible learning system for wrapping tables and lists in html documents. In: WWW-2002. (2002)

[11] Blei, D., Bagnell, D., McCallum, A.: Learning with scope, with application to information extraction and classification. In: UAI-2002. (2002)
[12] Lefferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML-2001. (2001)
[13] Cohen, W., McCallum, A., Quass, D.: Learning to understand the web. IEEE Data Engineering Bulletin **23** (2000) 17–24