

A survey of ontology-based web annotation

Naing, Myo Myo; Lim, Ee Peng; Goh, Dion Hoe-Lian

2003

Naing, M. M., Lim, E. P., & Goh, D. H. L. (2003). A Survey of Ontology-Based Web Annotation. Proceedings of the International Conference on Computer Applications, (Yangon, Myanmar), 113-123.

<https://hdl.handle.net/10356/93693>

© 2003 International Conference on Computer Applications. This is the author created version of a work that has been peer reviewed and accepted for publication by International Conference on Computer Applications. It incorporates referee's comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The official conference website is: <http://www.icca.org.in/>.

Downloaded on 09 Apr 2024 09:52:49 SGT

Naing, M.M., Lim, E.P., and Goh, D.H. (2003). A survey of ontology-based Web annotation. *Proceedings of the 1st International Conference on Computer Applications*, (January 15-16, Yangon, Myanmar), 113-123.

A Survey of Ontology-Based Web Annotation [★]

Myo-Myo Naing, Ee-Peng Lim^{**}, and Dion Hoe-Lian Goh

Centre for Advanced Information Systems, School of Computer Engineering,
Nanyang Technological University, Nanyang Avenue, N4-B3C-13, Singapore 639798,
SINGAPORE

`mmnaing@gmail.ntu.edu.sg`, `aseplim@ntu.edu.sg`, `ashlgoh@ntu.edu.sg`

Abstract. Heterogeneity and a lack of central control over the organization of Web information have led to great difficulties in processing them automatically. To overcome the difficulties, one attempts to add metadata to Web pages and this is known as Web Annotation. Ontology-based Web Annotation is a recent but promising area which enhances conventional Web Annotations by incorporating ontology with pre-defined semantic structures. The resulting annotations are useful not only for navigating Web information from diverse sources but also for improving information retrieval and classification. In this paper, we survey the different Ontology-based Web annotation methods in order to provide an organized view on the existing state-of-the-art, to identify potential areas of research, and to discuss promising ways of enhancing Ontology-based Web Annotation.

Keywords: Ontology, Web Annotation, Metadata

1 Introduction

1.1 Background and Motivation

The World Wide Web (WWW) provides abundant information for sharing and analysis. As the Web consists of large number of Web sites operated by autonomous organizations providing a wide range of Web content, locating useful information on the Web can be a daunting task. Web sites and their Web pages are often structured in different styles and formats making them more eye-pleasing but also harder for their content to be automatically extracted and processed. Without a good semantic knowledge about the Web sites and pages, it is also difficult to build sophisticated applications around the Web.

In this paper, we focus on the creation of metadata for the Web also known as **Web Annotation**[24]. Web Annotations can be defined in many different ways. In the more narrow definition, Web Annotations can be any comments, notes,

[★] This work is partially supported by the SingAREN21 research grant M48020004.

^{**} Dr. Ee-Peng Lim is currently on leave at Dept. of SEEM, Chinese University of Hong Kong, China.

explanations, references, examples, advices, correction or any other type of external remarks that can be attached to the whole or part of Web pages without modifications to the original pages. More broadly, Web Annotation can refer to any remarks that can be attached to any Web entities, including Web sites, Web pages and their hyperlinks. Examples of Web Annotation systems are ComMentor¹, Annotator², Third Voice³, CritLink⁴, CoNote⁵ and Futplex⁶ (Please see [11] for more details). These Web annotation systems support creation, storing and retrieval of annotation information. However, the Web Annotations created using these systems may not conform to some common semantic structures pertinent to the underlying domains of the Web content.

Studies on the Web Annotations have concluded that annotation without a controlled and well-defined model cannot reap any intended benefits because this just would build another layer of data which share the same heterogeneity and autonomy characteristics with the underlying Web [1]. Thus, attempts have been made to create Web Annotations based on some well-defined semantic structures or models, known as **ontologies** [12, 7]. To distinguish such annotation approach from the rest, we call this the **Ontology-based Web Annotation**(OWA).

1.2 Objectives

This paper provides a survey on *Ontology-based Web Annotations Methods*, referring to the techniques of semi-automatically creating annotations for Web objects using one or more ontology. We survey the existing Ontology-based Web Annotation techniques in order to achieve the following objectives:

1. To determine the features required for designing the Ontology-based Web Annotation Systems.
2. To compare different Ontology-based Web Annotation methods to assist potential users in selecting an approach that meets their needs.

Based on our findings, we describe the limitations of current methods and discuss promising ways to enhance Ontology-based Web Annotation.

1.3 Paper Outline

The remainder of the paper is organized as follows. In Section 2, we formally define the Ontology-based Web Annotation problem. Section 3 provides the various methods used for Ontology-based Annotation. Finally in section 4, we draw conclusions on the existing state-of-the-art and propose the additional requirements for Ontology-based Web Annotation as future work.

¹ <http://hci.stanford.edu/commentor/>

² <http://www.foresight.org/WebEnhance/Annotate.html>

³ <http://www.thirdvoice.com/>

⁴ <http://crit.org/http://crit.org/critlink.html>

⁵ <http://www.cs.cornell.edu/home/dph/annotation/annotations.html>

⁶ <http://gewis.win.tue.nl/applications/futplex/>

2 Ontology-based Web Annotation

2.1 Formal Definition of Ontology

For Web Annotation, ontology serves as the metadata schema for organizing semantically interrelated data on the Web. In the following, we give the formal definition of ontology.

Definition 1. (Ontology) *An ontology consists of 6 elements $\{C, A^C, R, A^R, H, X\}$, where C represents a set of concepts; A^C represents a collection of attribute sets, one for each concept; R represents a set of relationships; A^R represents a collection of attribute sets, one for each relationship; H represents a concept hierarchy; and X represents a set of axioms.*

Each concept c_i in C represents a set of objects of the same kind, and can be described by the same set of attributes denoted by $A^C(c_i)$. Each relationship $r_i(c_p, c_q)$ in R represents a binary association between concepts c_p and c_q , and the instances of such a relationship are pairs of (c_p, c_q) concept objects. The attributes of r_i can be denoted by $A^R(r_i)$. H is a concept hierarchy derived from C and it is a set of parent-child (or superclass-subclass) relations between concepts in C . $H(c_p, c_q)$ if c_q is a subclass, or sub-concept, of c_p . Each axiom in X is a constraint on the concept's and relationship's attribute values or a constraint on the relationships between concept objects. Each constraint can be expressed like a prolog like rule.

Note that the above ontology definition is very similar to that of conceptual schema in traditional databases. However, there are several key differences:

- The concept and relationship attributes are not mandatory items for the concepts and relationships. In other words, it is possible to have some concept attributes not specified for some of the concept objects. Similarly, one may not find values existing in the all attributes for a set of instances of the same relationship. The database concept schema requires however every object instance to contain attribute values in all their attribute.
- The domain of concept and relationship attributes are not fixed. In a database's conceptual schema, it is implicit that each attribute has a fixed domain of values. Such requirement does not exist in ontologies.
- In a database conceptual schema, each concept is required to have some attribute(s) serving as the key of the concept objects. In ontology, the key is not required. An implicit identifier is instead adopted for every concept object and relationship object.

In the following, we describe a part of simple University ontology that will be used in the rest of this paper.

Example 1. $Univ_Ontology = \{C_{univ}, A_{univ}^C, R_{univ}, A_{univ}^R, H_{univ}, X_{univ}\}$ where

- $C_{univ} = \{Student, PhDStudent, AcademicStaff, Professor, Department, Course, Project\}$

- $A_{univ}^C(Student) = \{name, matricnum, age, email\}$
 $A_{univ}^C(PhDStudent) = \{name, matricnum, age, email, project, supervisor\}$
 $A_{univ}^C(AcademicStaff) = \{name, title, staffid, email, url\}$
 $A_{univ}^C(Professor) = \{name, staffid, email, url, affiliation\}$
 $A_{univ}^C(Department) = \{name, researcharea, location, phone, url\}$
 $A_{univ}^C(Course) = \{name, title, reference, period\}$
 $A_{univ}^C(Project) = \{title, sponsor, researchfield, people, url, publication\}$
- $R_{univ} = \{Supervise(Professor, PhDStudent), WorkIn(Professor, Department),$
 $Major(Student, Department), Teach(Professor, Course),$
 $TaughtBy(Course, Professor), Faculty(AcademicStaff, Department),$
 $Take(Student, Course), Memberof(Student, Project)\}$
- $A_{univ}^R(Supervise) = \{startdate, enddate\}$
 $A_{univ}^R(WorkIn) = \{apptdate\}$
 $A_{univ}^R(Major) = \{academicyear\}$
 $A_{univ}^R(Teach) = \{semester, year\}$
 $A_{univ}^R(Take) = \{semester, year\}$
- $H_{univ} = \{(Student, PhDStudent), (AcademicStaff, Professor)\}$
- $X_{univ} = \{inverse(TaughtBy(Course, Professor) \leftarrow Teach(Professor, Course),$
 $transitive(Memberof(Student, Project) \wedge Subconcept(Student, PhDStudent)$
 $\rightarrow Memberof(PhDStudent, Project))\}$

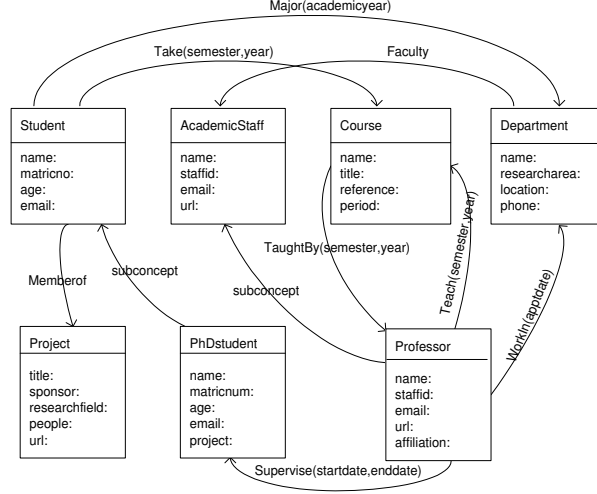


Fig. 1. An example part of University Ontology

2.2 Formal Definition of the Ontology-based Web Annotation Problem

Formally, we define the Ontology-based Web Annotation problem as follows:

Definition 2. (Ontology-based Web Annotation Problem) *Given a set of ontologies $O = \{O_1, O_2, \dots, O_n\}$ where $O_i = \{C_i, A_i^C, R_i, A_i^R, H_i, X_i\}$ and a set of annotated items $I = \{i_1, i_2, \dots, i_m\}$, the Ontology-based Web Annotation problem consists of four sub-tasks:*

- Task 1: Assignment of annotated items to concepts
The assignment can be represented by a set of mapping functions, one for each ontology. We denote the assignment for ontology O_i by a mapping function $f_i : I \rightarrow C_i$.
- Task 2: Assignment of annotated item pairs to relationships
The assignment can be represented by a set of mapping functions, one for each ontology. We denote the assignment for ontology O_i by a mapping function $g_i : I \times I \rightarrow R_i$. Note that $g_i(i_j, i_k) = r_i(c_s, c_t)$ only if $f_i(i_j) = c_s$, and $f_i(i_k) = c_t$. In other words, the assignment of annotated item pairs to relationships must be consistent with the assignment of their pair members to concepts.
- Task 3: Assignment of literal strings to concept attributes
Let Σ be an alphabet of characters, and Σ^ denotes the set of all possible strings constructed from Σ . The assignment of literal strings to the concept attributes can be represented by a mapping function $h_i : I \times A_i^C \rightarrow \Sigma^*$. Note that $h_i(i_j, c_t.a) = w$ only if $f_i(i_j) = c_t$ and $a \in A_i^C(c_t)$.*
- Task 4: Assignment of literal strings to relationship attributes
The assignment of literal strings to the relationship attributes can be represented by a mapping function $l_i : I \times I \times A_i^R \rightarrow \Sigma^$. Note that $l_i(i_j, i_k, r_i.b) = w$ only if $g_i(i_j, i_k) = r_i(c_s, c_t)$ and $b \in A_i^R(r_i)$.*

Using the university ontology example, we illustrate the Ontology-based Web Annotation results as follows:

Example 2. Let O_1 be the *Univ_Ontology*, i_1, i_2, i_3, i_4 be the Web pages of Professor Ee-Peng Lim, his student Myo Myo Naing, the Software Department, and a research centre known as CAIS, respectively.

- Task 1: Assignment of annotated items to concepts and their attributes
 $f_1(i_1) = \text{Professor}$
 $f_1(i_2) = \text{PhDStudent}$
 $f_1(i_3) = \text{Department}$
 $f_1(i_4) = \text{ResearchCentre}$
 The meaning of the above assignments is self-explanatory.
- Task 2: Assignment of annotated item pairs to relationships
 $g_1(i_1, i_2) = \text{Supervise}(\text{Professor}, \text{PhDStudent})$
 $g_1(i_1, i_3) = \text{WorkIn}(\text{Professor}, \text{Department})$
 The assignment $g_1(i_1, i_3)$ indicates that the annotated item i_1 is related to annotated item i_3 by the *WorkIn* relationship.

- *Task 3: Assignment of literal strings to concept attributes*
 $h_1(i_1, \text{Professor.url}) = \text{"http://www.cais.ntu.edu.sg:8000/~aseplim/"}$
 $h_1(i_2, \text{PhDStudent.matricnum}) = \text{"G0001393F"}$
 The above assignments provide the URL attribute value to i_1 and the matriculation number attribute value to i_2 .
- *Task 4: Assignment of literal strings to relationship attributes*
 $l_1(i_1, i_2, \text{Supervise.startdate}) = \text{"14/06/2001"}$
 $l_1(i_1, i_2, \text{Supervise.enddate}) = \text{"14/06/2004"}$
 The above assignments provide the start and end date attribute values to the relationship instance between i_1 and i_2 .

3 Methods for Ontology-based Web Annotation

Generally, an Ontology-based Web Annotation method can be manual, semi-automatic or automatic. A manual annotation method requires the annotator to manually insert annotations into the annotated document or some annotation database. To reduce typographical errors and time required for manual annotation, graphical tools can be introduced [12, 21, 22, 7].

In using a graphical annotation tool, the annotator must first select the Web page to be annotated and the appropriate ontology. The annotator can then identify the Web object within the Web page to be annotated, browse the concepts and relationships in the chosen ontology, and create the necessary annotation. Creating annotations using graphical tool frees the annotators from familiarizing the ontologies prior to annotating Web objects. In spite of these tools, our survey has shown that most Ontology-based Web Annotation methods are manual [7, 12, 22, 21].

The manual approach is expensive and time consuming when annotating large number of Web objects. To address these limitations, semi-automatic or automatic annotation methods are highly desirable.

To realise semi-automatic and automatic ontology based Web Annotation methods, different knowledge discovery techniques can be used. In particular, the classification techniques can be used to classify Web pages and other Web objects into different categories. These categories may correspond to concepts and relationships in the ontologies. The classification techniques usually require a set of pre-classified (or labeled) data also known as the training data. These training data correspond to example concept and relationship instances. Once the classifiers that correctly assign these example instances to concepts and relationships are created, they can be used to assign the other Web objects to concepts and relationships accordingly.

Hence, the tasks of assigning concepts to Web objects and relationships to pairs of Web objects can be automated using classification techniques. While the accuracy of classification techniques may not be perfect, some human efforts can still be deployed to make the final corrections to the assignments made by the classifiers. Note that such correction efforts will be much less than in a pure manual approach.

The assignments of literal strings to concept and relationship attributes are the two other tasks in Ontology-based Web Annotation. These two tasks can be aided by information extraction techniques.

In this section, we will discuss these classification and information extraction approaches in more detail.

3.1 Classification approaches for Assigning Web Objects to Concepts

The assignment of Web objects to concepts is very much similar to the text classification problem especially when the Web objects to be assigned are Web pages. A survey of text classification techniques has been given in [18].

In text classification, documents to be classified are represented either as *set-of-words* or *bag-of-words*. The set-of-words representation indicates only word presence or absence in the document, while bag-of-words representation represents the document by the frequencies of words in the document [5]. Hence, 0-1 or word frequency vector can be used to represent the same document depending on the type of document representation.

In this section, we give an overview of two classification approaches.

Probabilistic Classification approach In the probabilistic classification approach, probabilistic classifiers are used. The classifiers must be constructed by using the training data consisting of documents already classified by some expert in a particular domain. The probabilistic model of each category is built using these training data.

The probabilistic model to predict the probability of a document belonging to a class is built by using well known Bayes' theorem:

$$P(c_j|\vec{d}_i) = \frac{P(c_j)P(\vec{d}_i|c_j)}{P(d_i)} \quad (1)$$

where $P(c_j)$ is the probability that a randomly selected document belongs to c_j and $P(\vec{d}_i)$ is the probability that a randomly selected document has vector \vec{d}_i as its representation. $P(\vec{d}_i|c_j)$ can be estimated with the assumption that any two words of a document vector are statistically independent of each other as follows.

$$P(\vec{d}_i|c_j) = \prod_{k=1}^n P(w_{ki}|c_j) \quad (2)$$

The probabilistic classification technique using such an assumption is known as the Naive Bayes technique [14]. Naive Bayes technique is currently one of the most widely used classification techniques.

The disadvantages of probabilistic classifiers are that their nature is numerical and a lot of calculations needed for classification. In addition they are not easily interpreted by humans.

Decision Tree Classification Approach In a decision tree classification approach, one or more decision trees are constructed to classify documents based on their term (word) features. Two kinds of nodes can be found in a decision tree, namely internal nodes and leaf nodes. Each internal node represents a term and each leaf node represents a class (i.e., either the document belongs to a category or not). The edges between nodes are labeled with *test conditions* each indicating a range of term weights in the document. A decision tree example is given in Figure 2. To classify a document using a decision tree, the term weights of the document are tested against the tree. Firstly, a test document will be tested for the existence of the term *research*. If the document consists of the term, it will be assigned to the *AcademicStaff* category. Otherwise, look for the next term *professor*. If that term exists in the document, it will be assigned to *AcademicStaff* category. If the document does not contain both *research* and *professor*, it will not be assigned to the *AcademicStaff* category, and it will be denoted by the *AcademicStaff*.

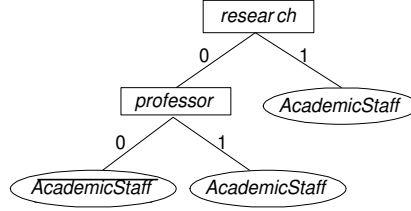


Fig. 2. An example Decision Tree

The document is first tested with the test condition at the root. Depending on the weight of the root term in the document, a specific edge is traversed. The classification step is performed recursively until a leaf node is reached. At that point, the document will either be classified under a category or rejected from the category.

The main challenge in decision tree classification is the construction of decision trees. The most widely used decision tree construction method based on divide-and-conquer strategy described in [18]. The learning method includes growing and pruning a decision trees based on the training data. Decision tree classifiers have been used for text categorization in [3].

3.2 Classification Approaches for Assigning Pairs of Web Objects to Relationships

To perform the task of assigning pairs of Web objects to relationships of an ontology, one can choose to treat each pair as an entity to be classified and borrow the classification techniques mentioned earlier. While this approach is

feasible, it does not explore the knowledge about the concepts already associated with the Web objects.

Very little research has been conducted on classifying pairs of Web objects. In this section, we present *relational path finding* technique.

Relational Path Finding Approach The basic idea of relational path finding approach is that a relational domain is viewed as a directed graph in which nodes represents the domain constants and edges correspond to relations between these constants. The relational path finding algorithm finds the small number of prototypical paths in this graph that connect the arguments/instances of the target relation.

Discovering of relationships between the documents in the WEB→KB project is based on recognizing the patterns represented by hyperlink paths among the documents [6, 4]. Rules representing the relationships between documents are able to recognize the instances under each relationship. Their algorithm is able to learn rule for paths consisting of more than one hyperlink. Detail description of their approach can be found in [4] and [5].

3.3 Information Extraction Approaches for Assigning Literal Strings to Attributes of Concepts and Relationships

Information extraction can be thought of as an activity of populating a structured database with information deriving from an unstructured or semi-structured data. To extract relevant information from documents, text extraction rules or patterns are the key components [19].

Text extraction rules are used to identify relevant information to be extracted from a document using some knowledge about the way the document is formatted or structured. In the case of HTML pages, this knowledge refers to the way HTML tags are used to markup their semantic content. Based on the text extraction rules, wrappers, software agents specially created to extract information using the extraction rules, can be used to assign the extracted information as attribute values of concept and relationship instances. The use of wrappers for information extraction has been described in [8, 13].

When the source documents do not carry any structural patterns for extraction, text extraction rules can still be constructed based on patterns involving syntactic relations between words or semantic classes of words. It requires syntactic analysis, semantic tagging, recognizers for domain objects such as professor or student names and the process to carry out inferences across sentence boundaries.

Information extraction approach is used to extract small fragments of text embedded in the pages [4] in WEB→KB project. The project has developed an information extraction learning algorithm called SRV [10] (“Sequential Rules with Validation”) based on the relational structure of the document. Other research efforts focused on information extraction from semi-structure online documents are WHISK [19], RAPIER [2], WIEN [13], Stalker [15], etc. and the comparison of the extraction patterns of these systems can be found in [16].

Recently, KA2 Community [9] and CREAM [17] proposed semi-automatic annotation methods based on information extraction approaches. Similar Ontology-based Annotation Tool was proposed by Knowledge Media Institute(KMi) [23].

With the rapidly increasing amount of online Web documents, research on information extraction still required to be investigated and the promising approaches can be somehow useful for automatic assignment of literal strings to the attributes of ontological elements.

4 Conclusion

Ontology-based Web Annotation enhances various application areas by deriving metadata from heterogeneous and distributed Web information. With OWA, users are able to access the Web with uniform and integrated view, get more relevant results, navigating only on interested categorized sites and achieve the semantic relations between documents without blindly wandering through the Web.

We have learnt that most of the Ontology-based Web Annotation systems are developed as a front end for enhanced query processing. Our survey has shown that almost all annotation methods used in existing systems are manual while very few focus on semi-automatic annotation, indicated that this is currently an active area of research. We also highlight that different classification and information extraction techniques can be applied for semi-automatic OWA tasks. Research in the area of automatic OWA methods is nascent and there are very few(if any) systems employing such an approach. To benefit from OWA in a large scale, it is therefore important to pursue automated OWA methods in the future by enhancing different classification and information extraction techniques.

References

1. M. Bremer and M. Gertz. Web data indexing through external semantic-carrying annotations. In *Proc. of the 11th Int. Workshop on Research Issues in Data Engineering: Document Management for Data Intensive Business and Scientific Applications*, pages 69–76, Heidelberg, Germany, April 2001.
2. M. E. Califf and R. J. Mooney. Relational Learning of Pattern-Match Rules for Information Extraction. In *AAAI Spring Symp. on Applying Machine Learning to Discourse Processing*, pages 6–11, Menlo Park, CA, 1998. AAAI Press.
3. W. W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems*, 17(2):141–173, 1999.
4. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. M. Mitchell, K. Nigam, and S. Slattery. Learning to Construct Knowledge Bases from the World Wide Web. *Artificial Intelligence*, 118(1-2):69–113, 2000.
5. M. Craven and S. Slattery. Relational Learning with Statistical Predicate Invention: Better Models for Hypertext. *Machine Learning*, 43(1/2):97–119, 2001.
6. M. Craven, S. Slattery, and K. Nigam. First-Order Learning for Web Mining. In *European Conf. on Machine Learning*, pages 250–255, Chemnitz, Germany, 1998.

7. S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology based access to distributed and semi-structured information. In *Proc. of the Semantic Issues in Multimedia Systems Conference*, Boston, MA, USA, 1999.
8. Line Eikvil. Information Extraction from World Wide Web-A Survey. Technical Report 945, Norweigan Computing Center, 1999.
9. M. Erdmann, A. Maedche, H. P. Schnurr, and S. Staab. From Manual to Semi-automatic Semantic Annotation: About Ontology-based Text Annotation Tools. *Computer and Information Science*, 6, 2001.
10. D. Freitag. Information Extraction from HTML: Application of a General Machine Learning Approach. In *Proc. of 15th Int. Conf. on AI*, pages 517–523, Madison, WI, 1998.
11. R. M. Heck, S. M. Luebke, and C. H. Obermark. A Survey of Web Annotation Systems, 1999. http://www.math.grin.edu/~rebelsky/Research/Summer1999/Papers/survey_paper.html.
12. J. D. Heflin. *Towards the Semantic WEB: Knowledge Representation in a Dynamic, Distributed Environment*. PhD thesis, University of Maryland, 2001.
13. N. Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118(1-2):15–68, 2000.
14. T. Mitchell. *Machine Learning*. McGraw Hill, 1997.
15. I. Muslea, S. Minton, and C. Knoblock. A Hierarchical Approach to Wrapper Induction. In Oren Etzioni, Jörg P. Müller, and Jeffrey M. Bradshaw, editors, *Proc. of the 3rd Int. Con. on Autonomous Agents (Agents'99)*, pages 190–197, Seattle, WA, USA, 1999. ACM Press.
16. Ion Muslea. Extraction patterns for information extraction tasks: A survey. In *In AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999.
17. S. Staab S. Handschuh. Authoring and Annotation of Web Pages in CREAM. In *Proc. of the 11th Int. WWW Conf.*, Honolulu, Hawaii, USA, May 2002.
18. F. Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Survey*, 34(1):1–47, March 2002.
19. Stephen Soderland. Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning*, 34(1-3):233–272, 1999.
20. S. Staab, A. Maedche, and S. Handschuh. An Annotation Framework for the Semantic Web. In *Proc. of the 1st Workshop on Multimedia Annotation*, Tokyo, Japan, 2001.
21. S. Staab, A. Mdche, and S. Handschuh. Creating Metadata for the Semantic Web—An Annotation Environment and the Human Factor. [cite-seer.nj.nec.com/378985.html](http://citeseer.nj.nec.com/378985.html).
22. S. H. Steffen. CREAM- Creating Relational Metadata with a Component-based, Ontology-driven annotation Framework. In *Proc. of the 1st Int. Conf. on Knowledge Capture*, Canada, 2001.
23. M. V.-Vera, E. Motta, J. Domingue, S. Shum, and M. Lanzoni. Knowledge Extraction by using an Ontology-based Annotation Tool. In *Proc. of the Workshop Knowledge Markup and Semantic Annotation*, Victoria, Canada, October 2001.
24. R. Zohar. Web Annotation-an Overview, February 1999. <http://www-ee.technion.ac.il/~ronz/annotation/>.