Project Number:	**215219**
Project Acronym:	**SOA4ALL**
Project Title:	**Service Oriented Architectures for All**
Instrument:	**Integrated Project**
Thematic Priority:	**Information and Communication Technologies**

D3.3.1 Ontology Instantiation State of the Art Report

Activity N:	2 - Core Research and Development	
Work Package:	3 - Service Annotation and Reasoning	
Due Date:		30/08/2008
Submission Date:		18/07/2008
Start Date of Project:		01/03/2008
Duration of Project:		36 Months
Organisation Responsible of Deliverable:		UKARL
Revision:		1.4
Author(s):	Maria Maleshkova (UKARL) Iván Martínez (ISOCO)	

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)

Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission)	
RE	Restricted to a group specified by the consortium (including the Commission)	
CO	Confidential, only for members of the consortium (including the Commission)	

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
1.0	18.07.2008	Initial Version	Maria Maleshkova (UKARL)
1.1	28.07.2008	Update Version	Iván Martínez (ISOCO)
1.1	06.08.2008	Review of the deliverable in version 1.1	Jürgen Vogel (SAP)
1.1	06.08.2008	Review of the deliverable in version 1.1	Pierre Grenon (OU)
1.2	07.08.2008	Deliverable changed based on the comments of the reviewers	Maria Maleshkova (UKARL)
1.3	27.08.2008	Deliverable changed based on the comments of the reviewers	Iván Martínez (ISOCO)
1.4	29.08.2008	Deliverable changed based on input from OU	Maria Maleshkova (UKARL)

Table of Contents

List of Figures

List of Tables

Glossary of Acronyms

Acronym	Definition
AAT	Art and Architecture Thesaurus
D	Deliverable
EC	European Commission
FOL	First Order Logic
HMM	Hidden Markov Model
IDF	Inversed document frequency
IF	Information Filtering
IR	Information Retrieval
ILP	Inductive logic programming
LEILA	Learning to Extract Information by Linguistic Analysis
ML	Machine learning
MOAT	Meaning Of A Tag
NB	Naive Bayes
NLP	Natural language processing
NP	Noun phrase
RSS	Rich Site Summary
SCOT	Social Semantic Cloud of Tags ontology
TF	Term frequency
WP	Work Package

Executive summary

This deliverable ***Ontology Instantiation State of the Art Report*** presents an overview and evaluation of the most relevant methods, techniques and tools used for the task of ontology learning. Ontology learning is the process of acquiring (constructing or integrating) an ontology and it plays a major role for the creation of semantic Web service descriptions in the SOA4ALL project. The choosing of the proper method for ontology learning will ensure the successful completion of the tasks envisioned in work package 3, since the annotation of services is highly dependent on the ontology, on which these annotations are based. This process of ontology learning can be semi- or fully automatic.

Ontologies are an important mean of knowledge sharing and reuse. They have many fields of application, including web portals and communities, corporate knowledge management and semantic search and in the context of the SOA4ALL for the semantic description of Web services. A common understanding of the term *Ontology* is as a container for capturing semantic information of a particular domain. Another widely accepted definition of ontology in information technology and AI community is that of "a formal explicit specification of a shared conceptualization" (Gruber, 1994), where “formal implies that the ontology should be machine-readable, and shared that it is accepted by a group or community” (Buitelaar et al, 2005). As described in detail in this document, acquisition of ontologies can be performed through three major approaches:

1. Through composing an ontology from scratch, or by extending an existing ontology. This approach is one of the most suitable ones to be used in WP3.
2. Through the integration of existing ontologies.
3. Through the specification of a generic ontology and subsequently adapting it to a specific domain. This approach can enable the adaption and improvement of already developed ontologies to the goals of SOA4ALL.

The tasks involved in the process of ontology learning can also be classified into five major groups. Each of these tasks is described in detail in this deliverable and the different areas of application are addressed:

1. Concept identification. Concepts form the main building elements of an ontology.
2. Term identification. Terms are the symbolical representation of concepts and relations in an ontology.
3. Synonym identification. Synonyms are terms, which represent the same real object or event.
4. Relation identification: There are two types of relations, which can be identified: hierarchical and non-hierarchical. Hierarchical relations are semantic associations between two ontological concepts and organize concepts into a taxonomy.
5. Rules acquisition. Rules formalize constraints over the concepts and relations of an ontology.

The goal of this document is to provide an overview of the state of the art in ontology learning by presenting the different research and achievements based on the identified approaches, tasks and activities. After this, an overview of a comparative analysis of existing tools and approaches is given, in order to provide insights the implementation perspective of ontology learning. The comparative analysis is based on a set of predefined dimensions, which are used to compare existing tools and approaches, giving special emphasis to the ontology learning aspects that are of particular interest to the SOA4ALL project. Finally, some conclusions are made, which try to relate the analyzed state of the art with the tasks and goals pursued in WP3 in SOA4ALL. These conclusions provide recommendations

concerning the methodologies and implementations for the annotation of Web services developed in WP3 in SOA4ALL. In particular, this involves suggestions about the required input information, the learning approach itself, the level of automation and the resulting ontology. These recommendations can be summarized as follows:

- The input information has a semi-structured nature and should be complemented by patterns, which recognize terms based on the structure of the documents.
- The learning method should employ a pattern based approach, especially for the extraction of relations between concepts, in combination with machine learning. In addition, a domain expert should be involved, in order to improve the quality of the resulting ontology. Moreover, the developed method should enable the extraction of both concept and relation instances.
- The method used in the ontology learning process should take advantage of the tagging technology and should allow for a transparent capture of users' knowledge.

1. Introduction

Recently, ontologies have become increasingly important for the representation of semantic knowledge in a machine-readable manner. With the constantly growing amount of information, triggered by the popularity and the rapid growth of the Web, the importance of effective methods for information extraction and representation have increased. As a result, significant research effort has been invested in the development of practical information extraction solutions that have promised to ease the problem of the user's overload with information. Triggered by the difficulty and the challenges of extracting information from the Web, the movement to the Semantic Web was initiated. The goal is that the Semantic Web contains many more resources than the Web and that machine-readable semantic information is attached to all of these resources. The first steps towards this goal, is to tackle the problem of knowledge representation for all this semantic information, which is done by the development of ontologies.

Currently, research on the Semantic Web has focused on the development of domain or task-specific ontologies. In this way, after an ontology for a specific domain is provided, the next step would be to annotate semantically all related Web resources. However, if manually done, this process is very time-consuming and error-prone. Therefore, information extraction and data mining are the most promising solution for automating the annotation process. In the context of the SOA4ALL project, by collecting Web service descriptions and related documents and by applying information extraction and ontology learning methods, semantic Web service descriptions can be provided. In order to automate this process, a new methodology for ontology learning has to be developed, which combines information extraction and learning methods.

As part of the Web 2.0 movement, folksonomies are commonly used in the form of tagging systems, due to their ease of use. Folksonomies allow the integration of heterogeneous resources and the collaboration of users in the resource tagging process. However, it is difficult to work with such information because it has not any structure and it is user dependent. However, ontologies provide a framework to handle structured information and to draw conclusions from such structured information. In the context of the SOA4ALL project, the goal will be to extract structured information (ontologies) from knowledge built in a simple and collaborative way (folksonomies).

The collection of domain knowledge for constructing ontologies is a resource demanding and time consuming task. Thus, the automatic or semi-automatic construction, enrichment and adaptation of ontologies, is highly desired. In this section, we define the notions of ontology learning, ontology enrichment and ontology population, which will be used throughout this document. The process of automatic or semi-automatic construction, enrichment and adaptation of ontologies is known as ontology learning. Ontology learning also considers problems such as inconsistency resolution and ontology population. Ontology enrichment is the task of filling an existing ontology with additional concepts and placing them in the correct position of the taxonomy. Ontology population, on the other hand, is the task of adding new instances of concepts into the ontology. Finally, inconsistency resolution is the task of resolving inconsistencies that appear in an ontology with the goal to acquire a consistent resulting ontology.

The purpose of this deliverable is to provide an overview of the state of the art in ontology learning, by describing the major approaches and most important system implementations. All of the here discussed, systems and approaches are classified and discussed based on a number of criteria, including the ontology elements learned, the starting point, the learning approach and the final outcome. The goal is to determine the main approaches and their characteristics, which are suitable to be applied in WP3 of the SOA4ALL project.

1.1 Purpose and scope of this deliverable

The goal of this deliverable is to provide a critical overview of the state of the art in ontology learning and instantiation. One of the main tasks of WP3 is to develop a methodology for the extraction of semantic Web service annotations. In order to achieve this task, a suitable method for determining an ontology, which describes the information related to Web services, has to be developed. The ontology is going to be based on information extracted from service-describing documents collected from the Web. Therefore, an overview of the existing ontology learning approaches provides a valuable insight of the possible solutions and their advantages and disadvantages. A discussion of the already existing tools shows the implementation potential and the efficiency of some of the ontology learning methods.

1.2 Structure of the document

This document is divided into five main sections. Following the introduction, section 2 provides an overview of the types of elements learned and the associated tasks of the ontology learning process. For each of the various subtasks of ontology learning, the related research work is described. Section 3 describes the main approaches for ontology population and enrichment, as well as some of the other main activities contained in ontology learning. Section 4 provides a description of some of the major implementations done in the area of ontology learning, including important ontology population tools and ontology enrichment tools. Section 5 provides an overview of the existing tagging approaches and their potential in the context of SOA4ALL. Finally, sections 6 and 7 provide decision criteria for choosing the most suitable ontology learning approach and a conclusion.

2. Ontology learning tasks and learned elements

Ontologies are an important mean of knowledge sharing and reuse. They have many fields of application, including web portals and communities, corporate knowledge management and semantic search. In the context of SOA4ALL an ontology will be used for the semantic annotation on Web services. A common understanding of the term *Ontology* is as a container for capturing semantic information of a particular domain. The definition for ontology, which will be used in SOA4ALL is a machine-readable specification of a conceptualization of a domain (Gruber, 1994, Buitelaar et al, 2005).

Ontology learning is the process of acquiring an ontology. As described in detail in this document, acquisition of ontologies can be performed through three major approaches:

1. Through composing an ontology from scratch, or by extending an existing ontology (based on information extraction approaches applied in a specific domain). This approach is one of the most suitable ones to be used in work package 3.
2. Through the integration of existing ontologies. During the integration process, commonalities among ontologies that convey the same or similar domains are identified, in order to derive a new ontology. There is some research work already done in this area and three main approaches can be identified:
    a. Ontology alignment is based on establishing links between the separate ontologies and allowing them to reuse information from one another.
    b. Ontology merging results in a single coherent ontology.
    c. Ontology mapping is based on correspondences among elements of the separate ontologies
3. Through the specification of a generic ontology and subsequently adapting it to a specific domain. This approach can enable the adaption and improvement of already developed ontologies to the goals of SOA4ALL.

This deliverable explores mainly the construction of new ontologies and the creation of generic ontologies, which can be adapted to a specific domain. The automation of ontology construction plays a key role for achieving the objectives of work package 3 Service Annotation and Reasoning because the annotation of services can be effectively performed only if an ontology exists, which describes the information available about a service. In this way, given a learned service-describing ontology and a set of documents that describe one particular service, this service can be annotated by instantiating the ontology based on the given documents. Therefore, the approach used in WP3 is to create an ontology, whose instances for one service represent the semantic annotation of this service. The following section focuses on providing a brief overview of the challenges connected with the separate tasks of ontology learning, followed by a description of each of these tasks and related research work.

2.1 The challenge of ontology learning

This chapter presents an overview of ontology learning approaches based on a classification of the type of conceptual structure acquired as a result. The learned elements can be simply ontological knowledge or both lexical and ontological knowledge. The main lexical elements learned are terms and the main ontological elements are concepts, relations and rules.

Ontology learning comprises a whole domain of research topics, involving methods and techniques for the acquisition of an ontology from semantic information. Since ontology learning is closely related to the field of knowledge acquisition, a significant amount of the research work concentrates on the task of knowledge acquisition from text, through the re-use of widely adopted natural language processing and machine learning techniques. However, ontology learning is not the simple reuse of existing work and approaches under a different name because it adds some novel aspects to the problem of knowledge acquisition (Buitelaar et al, 2005):

- Interdisciplinary work: Since ontology learning is partially motivated by the Semantic Web, it involves interdisciplinary work and combines research from knowledge representation, logic, philosophy, databases, machine learning, natural language processing, image/audio/video analysis, etc.
- Heterogeneity of information: Ontology learning in the context of the Semantic Web must deal with heterogeneous data and thus improve existing approaches for knowledge acquisition, which targeted mostly small and homogeneous data collections.
- Evaluation: Much effort is being put into the development of extensive and rigorous evaluation methods in order to evaluate ontology learning approaches on well defined tasks with well-defined evaluation criteria.

The ontology learning process can be decomposed into 6 major tasks, forming a “layer cake” (Buitelaar et al, 2005). The complexity of the subtasks increases from the bottom to the top, starting with terms and building up to rules (Figure 1).

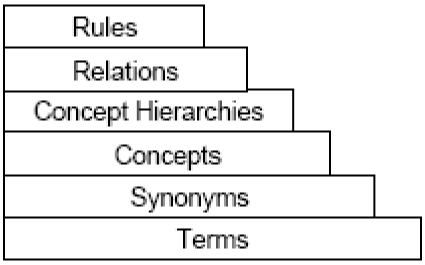


Figure 1: *Ontology learning “layer cake”.*

As already stated, the main goal of ontology learning is the definition of concepts and the relations between them. This requires knowledge of the involved concepts and relations and their “lexicalisations” or representations into objects of the real world. Therefore, the notation “term” is introduced, which to refer to these “lexicalisations” in order to reuse the terminology originating from the discipline of natural language processing. In addition, the knowledge about term synonyms must be also known: all terms that are synonyms refer to the same real object or event, and thus all lexicalise a single concept or relation. In addition, synonymy conflicts with the assumption that to each term corresponds a concept. This is important because without the identification of synonyms, redundant concepts or relations can exist in an ontology, which in most cases is undesired. Moreover, the acceptance of the principle of synonymy shows that concepts cannot be equated with terms and that ontologies are not simply a collection of terms.

After identifying the terms, synonyms and concepts, relations have to be determined. These are divided into two groups: hierarchical and non-hierarchical. Hierarchical relations are the relations that bring structure into the ontology, such as the “is-a” relation. On the other hand, non-hierarchical relations are all relations that are not used in the formation of the concept hierarchy.

Finally, an important property of an ontology is the usage of reasoning for deriving and making explicit facts that are implied by the knowledge contained in the ontology. These derivations are based on predefined rules, which represent the top and most complex layer of the ontology learning "layer cake". In the following subsections, the state of the art for each layer of this "cake" is briefly described.

2.2 Term Identification

The extraction of terms plays a major role for ontology learning. As defined in this deliverable, a term is an instance, which conveys a single meaning within a domain. If the corpus of relevant documents is only text, the notion of term would be equivalent to word. In this sense, a term can be words or phrases in textual corpora. The main objective of the term identification task, as part of the ontology learning, is the identification of terms that possibly convey concepts, which can be used for enriching an ontology. The term identification task can be decomposed into three subtasks (Krauthammer and Nenadic, 2004):

a. Term classification. During this task a semantic category is assigned to every recognized term. The term classification is important for the task of ontology learning because these categories are often the concepts of the domain. This is also the most relevant task of the term identification process in the context of work package 3.

b. Term recognition. During this task specific entities are identified and found in the data corpus.

c. Term mapping. During this task identified terms are linked with relevant entities in other data sources, such as vocabularies, lexica, thesauri and databases. This task is important for the identification of synonyms because it exploits similarities that potentially exist in the referred data sources, in order to identify clusters of terms that represent the same concept, i.e. synonyms.

The identification of terms is important not only for the discovery of concepts but also for textual information extraction and retrieval. Therefore, a number of approaches have been developed in research work. Among the most commonly used ones are the ones which use statistical methods. These approaches usually try to determine the significance of each word with respect to other words in a corpus, based on word occurrence frequencies. TF/IDF (Saltion et al., 1975) is usually employed for this task (Ahmad et al., 1994; Damerau, 1993), possibly combined with other methods such as latent semantic indexing (Fortuna et al., 2005) or taking into account co-occurrence information among phrases (Frantzi et al., 2000).

In addition to the statistical methods, clustering techniques are also commonly used for term identification. In this way, recognizable entities can be clustered into groups based on various similarity measures, where each cluster can be a possible term (comprised of synonyms). Other approaches (Kietz et al., 2000; Agirre et al., 2000; Faatz et al., 2002) combine clustering techniques and other resources like WWW and WordNet to successfully extract terms. Moreover, enhanced term identification approaches result from combining natural language processing techniques with both frequency and clustering based approaches (such as morphological analysis, part-of-speech tagging and syntactic analysis) because terms usually are noun phrases or obey specific part-of-speech patterns (Gupta et al., 2002; Haase and Stojanovic, 2005). Finally, morphological clues (such as prefixes and suffixes) can be extremely useful for some domains: suffixes like "-fil" and "-it is" quite often mark terms in medical domains (Haase et al, 2005; Haase and Volker, 2005).

Especially in the cases where the available corpus of data consist of text documents, most ontology learning systems (to mention few, TEXT-TO-ONTO; DODDLE II; Kietz, et al., 2000; Borgo, et al., 1997) use predefined lexicons for term identification, while some of them (HASTI; SYNDIKATE) learn lexical knowledge about words too.

After term classification and recognition techniques are applied on the given corpus of data, term mapping techniques can be applied in order to identify sets of synonyms. The following section descries the ontology learning task of synonym identification.

2.3 Synonym Identification

Synonyms are terms that refer to the same real object or event. In other word, synonyms represent term variants in a corpus that can be thought as depicting the same concept or relation. In this context, a significant amount of work has been performed, mainly for the text modality, by exploiting resources such as WordNet (Fellbaum, 1998). Other researchers employ standard word meaning disambiguation techniques (Lesk, 1986; Yarowsky, 1992; Dagan et al., 2005) and seek to identify the most appropriate (WordNet) meaning of each term in order to collect associated synonyms.

Other approaches for term synonyms location are based on clustering, mainly based on Harris' distributional hypothesis according to which similar terms in meaning tend to share syntactic contexts (Harris, 1968; Hindle, 1990; Lin and Pantel, 2001; Lin and Pantel, 2002). Related is also work performed in the field of information retrieval for term indexing, such as the family of Latent Semantic Indexing algorithms (LSI, LSA, PLSI, etc.). These apply dimension reduction techniques to reveal inherent relations between words, in order to form clusters of words (Schütze, 1993; Landauer and Dumais, 1997). Finally, more recent approaches try to extract synonyms by applying statistical approaches defined over the Web (Turney, 2001; Baroni and Bisi, 2004).

The following section discusses shortly the notion of concepts and focuses on describing techniques for identifying concepts, given a specific corpus.

2.4 Concept Identification

A concept can be anything about which something is said and can be abstract or concrete, elementary or composite, real or fictional, the description of a task, function, action, strategy, reasoning process, etc (Corcho & Gomez-Perez, 2000). It can be said that concepts are one of the most important constituents of an ontology. Still, what constitutes a concept is controversial. According to (Buitelaar et al, 2005) concept formation should provide:

- An intentional definition of the concept.
- A set of concept instances.
- A set of realizations (i.e. terms, words).

Concepts are represented by nodes in the ontology graphs and may be learned by the ontology learning system (such as HASTI; SYNDIKATE; Roux, et al, 2000; Soderland, et al., 1995). They may be extracted from input or be created during the ontology refinement from other concepts. In other words, they may have or not corresponding elements in the input.

Most research work related to concept identification is based on text modality. In this sense, two types of intentional concept definition can be identified: the informal and formal concept definition. An informal definition is a concept definition, which does not define a concept in terms of properties and relations between them, but in a more general, descriptive way. An example for an informal definition is a textual description or a concept gloss of a concept extracted from a dictionary. Informal concept identification is quite rare, with only one approach appearing in the literature, the OntoLearn system (Navigli&Velardi et al, 2005). This approach associates WordNet glosses with domain specific concepts.

On the other hand, formal concept definition builds on top of term and synonym identification, by formulating concepts as clusters of "related" terms. This is done by exploiting relations found among terms through approaches that will be described in the following subsections. After the concepts are identified, the acquisition of a set of instances for a concept is known as ontology population or ontology tagging and will be presented in greater detail in a subsequent chapter. Some ontology learning systems use an existing ontology and just populate it by instances of classes and relations (WEB→KB; Suryanto & Compton, 2001). Most of these systems do not learn new concepts and just learn instances of existing classes.

2.5 Conceptual Relations

After the concepts in an ontology are identified, relations between these concepts can also be determined. There are two major approaches for identifying conceptual relations. First, a relation is a node in the ontology, so it is a concept and may be learned like other concepts. Second, a relation relates two or more concepts and so it should be learned as a subset of a product of n concepts (for n>1).

Therefore, relations may be identified intentionally, independent to what concepts do they relate, or extensionally, considering the concepts, which are being related to each other. The first case will be counted in the first type of learned elements (concepts) and the second case in the second type (conceptual relations). In some systems both aspects of relations may be learned (Shamsfard and Barforoush, 2004), while in some others the relations themselves (the first aspect) are predefined and their occurrences (the second aspect) will be learned using some templates (Borgo, et al., 1997).

2.6 Taxonomy Construction

An important part of an ontology is its taxonomy, or the hierarchy of concepts. Taxonomies organize ontological knowledge using generalization/specialization relationship through which simple/multiple inheritance could be applied (Corcho & Gomez-Perez, 2000). Inclusion relations (also known as “is-a” relations) provide a tree view of the ontology and imply inheritance between super-concepts and sub-concepts. One common approach for textual domains is the use of lexico-syntactic patterns (such as Hearst patterns (Hearst, 1992)). In this approach syntactic elements, i.e. noun phrases, are combined with specific phrases to identify inclusion relations. Examples of such patterns can be the following ones (NP stands for noun phrase):

- NP such as NP, NP,..., and NP
- such NP as NP, NP,..., or NP
- NP, NP,..., and other NP
- NP, especially NP, NP,..., and NP
- NP is a NP

Several systems have been proposed based on simple variations of the above idea, such as (Morin, 1999; Kietz et al., 2000; Iwanska et al., 2000). More recent systems combine also machine learning and pattern learning algorithms to automate pattern construction (Downey et al., 2004; Snow et al., 2004; Agichtein and Gravano, 2000). For non-textual domains, machine learning such as hierarchical clustering can be used. Some details can be found in (Zavitsanos et al., 2006) and (Buitelaar et al, 2005).

2.7 Semantic Relation Extraction

Semantic relation or also non-taxonomic conceptual relations refer to any relation between concepts except the ISA relations, such as synonymy, meronymy, antonymy, attribute-of, possession, causality and other relations. These relations can be extracted with approaches similar to the ones for extracting taxonomic relations. In textual documents lexico-syntactic patterns play an important role, since verbs represent an action or a relation between recognizable entities in sentences (learned by systems such as HASTI; TEXT-TO-ONTO; Agirre et al, 2000; Gamallo, et al., 2002). As a result, verbs are assumed to express a relation between two entities. This assumption can be useful for enriching an ontology because in this way the involved entities can be associated with concepts from the ontology. Systems like RelExt tool (Schutz and Buitelaar, 2005) use such patterns to identify related pairs of concepts. Additionally, semantic clustering of verbs has been reported to help in situations where extraction of specific relation types is desired (Schulte im Walde, 2000). Finally, association rule mining algorithms have been used for the acquisition of non-taxonomic relations for ontology enrichment (Maedche and Staab, 2000a; Maedche and Staab, 2000b).

2.8 Rules acquisition

Rule acquisition is one of the least explored tasks in ontology learning. An initial attempt to formulate the problem is presented in (Lin and Pantel, 2001b), where an unsupervised method for discovering inference rules from text is presented. Learned rules are similar to (from Lin and Pantel, 2001b) "*X is author of Y ≈ X wrote Y*, *X solved Y ≈ X found a solution to Y*, and *X caused Y ≈ Y is triggered by X*". Relevant is also the field of inductive logic programming (ILP), where recent attempts have been made to address reasoning for the Semantic Web (Lisi, 2005).

2.9 Meta knowledge

Beside systems that learn ontological knowledge, there are systems which learn how to learn/extract ontological knowledge. They learn meta knowledge such as rules to extract instances, relations and specific fields from the Web (WEB→KB) or patterns to extract knowledge from text (Soderland, et al., 1995) or association rules in a corpus (Cherfi & Toussaint, 2002).

3. Learning Approaches

The selection of a suitable ontology learning approach plays a key role for achieving the objectives of WP3 Service Annotation and Reasoning because the annotation of services can be effectively performed only if an ontology exists, which describes the information available about a service. In this way, given a learned service-describing ontology and a set of documents that describe one particular service, this service can be annotated by instantiating the ontology based on the given documents. Therefore, the approach used in WP3 is to learn an ontology, whose instances for one service represent the semantic annotation of this service.The actual methodology used for ontology learning can be classified based on two main criteria. The first one is based on the type and form of input information and the second one considers the type of the approach used for the actual ontology learning. Each of the two classification criteria is described briefly in this section and related research work is discussed.

3.1 Input information

3.1.1 Starting Point

The starting point relates to determining from where to start ontology acquisition and from what to learn. Ontology learning systems rely on their background knowledge and acquire new knowledge elements based on their input. The quality and quantity of the prior knowledge and the type, structure and language of the input differ. However, this input information plays a major role for the quality and the efficiency of the ontology learning.

The background knowledge may be presented in linguistic (lexical, grammatical, templates, etc) or ontological (base ontology) resources. Many domains and approaches use a predefined lexicon used to process texts (such as Kietz et al, 2000). In some case the lexicon is a semantic lexicon covering ontological knowledge too (such as using (Euro) WordNet in TEXT-TO-ONTO; SYNDIKATE; DODDLE II; Agirre et al, 2000; Termier et al, 2001).

3.1.2 Input

Ontology learning systems aim to extract knowledge of interest from input corpus of documents. Input sources can be classified based on type and language.

1. Type: The type of input, from which the ontology learning system acquires knowledge:

   - Structured data: Database schemata, existing ontologies (Williams & Tsatsoulis, 2000), knowledge bases (Suryanto & Compton, 2000) and lexical semantic nets such as WordNet.

   - Semi structured data: Dictionaries systems, HTML and XML docs and DTD’s (document type definitions). Growing interest in the Semantic Web leads to increased interest in building ontologies for the Web. Therefore, there are a number of approaches targeted at learning ontologies from semi structured data in web documents. TEXT-TO-ONTO, WEB→KB, and (Kavalek & Svatek, 2002) are instances of such systems. In the context of SOA4ALL, the documents which can be used for the annotation of Web services will most probably by semi-structured (HTML and XML). Therefore, a particular attention should be paid to this group of ontology leaning systems.

   - Unstructured data: Natural language ontology learning, done by exploiting the interacting constraints on the various language levels, in order to discover new concepts and relationships between concepts (OLT’2002).

2. Language: The input may be natural language texts in English (DODDLE II; Wagner, 2000; Termier et al, 2001), German (SYNDIKATE; TEXT-TO-ONTO), French (ASIUM; SVETLAN'), or data presented in artificial languages such as XML (TEXT-TO-ONTO) or RDF.

3.1.3 Preprocessing

Finally, the preprocessing done on the input information, before applying the ontology learning techniques, also has an important impact on the resulting ontology and its properties. The preprocessing usually addresses the structure of the input information, since some structures are more suitable to learn from then others. The most commonly used preprocessing in learning ontologies from texts is the linguistic preprocessing. Most existing systems use text processing such as tokenizing, part-of-speech (POS) tagging, and syntactic analysis to extract essential structures from input texts.

3.2 Learning Methodology

The ontology learning approaches can be divided into two main groups, statistical or symbolic. The symbolic approaches include the logical, linguistic based and template driven approaches. In addition, heuristic methods may be used to facilitate each approach. There are also hybrid approaches, which combine two or more of the above approaches and employ their benefits and eliminate their limitations. For SOA4ALL a hybrid approach, which uses some statistical analysis in combination with linguistics and templates, will probably be the most suitable to use.

3.2.1 Statistical

The statistical approach uses statistical analysis performed on data gathered from the input. This approach is very common in ontology learning. As an example, WEB→KB uses a statistical bag-of-words approach to classify Web pages, (Wagner, 2000) exploits a modification of the algorithm by Li & Abe (1996) for acquisition of selectional preferences and locating concepts in the ontology at the appropriate generalization level. Also DODDLE II use statistical analysis of data to learn conceptual relations from texts. Hidden Markov Model (HMM) is also used (Bikel, et al., 1999) to find and label names and other numerical entities.

Another approach is the use of statistical indices (Cherfi & Toussaint, 2002) to rank the association rules that are more capable of reflecting the complex semantic relations between terms. One positive feature of statistical methods is that they can be applied both on isolated words or batches of words together. They can differ in size of batches, distribution function and statistical analysis done on the input data.

Models based on isolated words are often called unigram or bag-of-words models. These models ignore the sequence, in which the words occur. Since the unigram model naively assumes that the presence of each word in a document is conditionally independent of all other words in the document given its class, this approach, when used with Bayes Rule is often called naive Bayes (Craven, et al., 2000). WEB→KB's method for classifying web pages is naive Bayes, with minor modifications based on Kullback–Leibler Divergence.

On the other hand, some statistical methods often consider batches of words. The main idea common to these approaches is that the semantic identity of a word is reflected in its distribution over different contexts. In this way, the meaning of a word is represented in terms of words cooccurring with it and the frequencies of the cooccurrences (Maedche, et al., 2002). The occurrence of two or more words within a well-defined unit of information (sentence, document) is called a collocation (Heyer, et al., 2001). Learning by Collocations and Cooccurrences is the most commonly used method in statistical learning of ontological knowledge. During the process of learning by collocation and cooccurrence, first a collocation matrix is created. After this, based on the statistical analysis of this structure, the conceptual relations between concepts are discovered. One example is (Heyer, et al., 2001), where a kind of average context for every word W is formed by all collocations for W with a significance above a certain threshold.

3.2.2 Logical

Logical methods can also be used to extract ontological knowledge from a corpus of documents. Some examples include inductive logic programming (ILP), FOL (First Order Logic) based clustering, FOL rule learning (WEB→KB) and propositional learning (Bowers, et, al., 2000). Logic-based learning methods may discover new knowledge by deduction or induction. Deduction based learning systems exploit logical deduction and inference rules such as resolution. On the other hand, induction-based learning systems (such as WEB→KB; Bowers, et, al., 2000) induce hypotheses from observations and synthesize new knowledge from experience.

Inductive Logic Programming (ILP) is a combination of inductive learning and logic programming, where the hypotheses and observations are represented in first order logic or variants of it (Muggleton & De Raedt, 1994). FOIL (Quinlen & Cameron-Jones, 1993) is one of the best-known and most successful empirical ILP systems and some ontology learning systems already use variants of it. FOIL is a greedy covering algorithm that induces concept definitions represented as function-free Horn clauses, optionally containing negated body literals. It induces each Horn clause by beginning with an empty tail and using a hill-climbing search to add literals to the tail until the clause covers only (mostly) positive instances.

3.2.3 Linguistic

Linguistic approaches include syntactic analysis (ASIUM), morpho-syntactic analysis (Assadi, 1997), lexico-syntactic pattern parsing, semantic processing (HASTI) and text understanding (SYNDIKATE). These approaches are used to extract ontological knowledge from natural language texts. Linguistic approaches, because of their nature, are usually language dependent and perform the preprocessing on the input text to extract essential knowledge to build ontologies from texts. For example, Assadi (1997) performs a partial morpho-syntactic analysis to extract “candidate terms” from technical texts. The result of the morpho-syntactic analysis would be a network of noun phrases, which are likely to be terminological units. Any complex term is recursively broken up into two parts: head and expansion, which are both linked to the complex candidate term in a terminological network. The network will then be used by the conceptual analyzer to build a classification tree.

Another example for a linguistic approach is SYNDIKATE, which uses text understanding techniques to acquire knowledge from real-world texts. It integrates requirements from the analysis of single sentences, as well as those of referentially linked sentences forming cohesive texts. The results of the syntactic analysis are described in a dependency graph, in which nodes are words and edges are dependency relations such as specifier, subject, dir-object, etc. After the graph is created, a semantic interpretation is performed, which identifies conceptual relations in the knowledge base between conceptual correlates of words.

Finally, another linguistic method is Lexico-syntactic pattern parsing. In this method the text is scanned for predefined lexico-syntactic patterns which can indicate a relation of interest, e.g. the taxonomic relation (Maedche, et, al., 2002). The following section discusses the learning of ontologies based on the extraction of such patterns.

3.2.4 Pattern Based / Template Driven

Pattern based and template drive approaches are commonly used in the information extraction field and are, therefore, also applied to the ontology learning domain. In template driven methods the input is searched for predefined keywords, templates or patterns, which specify given relations. There are different types of templates, depending on the ontology elements, which they are targeted at extracting. Some common types of temples are syntactic or semantic and general or special purpose. One major research milestone in pattern matching is given by Hearst (1992). He introduced lexico-syntactic patterns in the form of regular expressions, which serve for the extraction of hyponymy/ hyperonymy relations from texts.

Related to semantic template are the symbolic interpretation rules in (Gamallo, et al., 2002). They use grammatical patterns to map syntactic dependencies onto the semantic relations such as hyperonymy, possession, location, modality, causality and agentivity. Another work is done by Sundblad (2002) in which some linguistic patterns are used to extract hyponymy and meronymy relations from question corpora. Heyer, et al., (2001) proposed patterns to extract first names and instance-of relations from sentences

The patterns may be general and application/domain neutral, such as those proposed by Hearst, HASTI and Sundblad or specific to a domain or application such as those used by Assadi (1999) to extract knowledge from electric network planning texts. On the other hand patterns may be manually defined (HASTI; Sundblad, 2002; Gamallo, et al., 2002) or may be extracted (semi) automatically such as in PROMETHEE (Finkelstein-Landau & Morin, 1999), AutoSlog-TG (Riloff, 1996) and CRYSTAL (Soderland, et al., 1995). In the context of SOA4ALL domain specific patterns should be developed, which are especially targeted at detecting information characterizing Web services. Moreover, the initial approach for pattern provision can be manual. If the results of the ontology learning based on patterns are promising, a (semi-)automatic approach can be subsequently developed.

3.2.5 Heuristic Driven (Ad Hoc Methods)

Heuristic driven methods are not independent and complete methods; therefore, they should rather be used to support other approaches. For example, some of the methodologies based on heuristic driven approaches include TEXT-TO-ONTO; HASTI; InfoSleuth (Hwang, 1999) and (Gamallo, et al., 2002).

TEXT-TO-ONTO uses heuristic rules to increase the recall of the linguistic dependency relations. One of the used heuristics is the NP-PP-heuristic, which attaches all prepositional phrases to adjacent noun phrases. Another heuristic is the sentence-heuristic, which relates all concepts contained in one sentence, if other criteria fail. Finally, there is the title-heuristic, which links the concepts in the HTML title tags with all the concepts contained in the overall document.

HASTI uses some simplifying heuristics, in order to decrease the size of hypothesis space. For example, the priority-assignment heuristics is used to assign priorities to ambitious terms and candidate-choosing heuristics is used to choose a merge set in the ontology refinement task.

3.2.6 Multi Strategy learning

Multi strategy learning is applied in most systems, which learn more than one type of ontology elements. They apply combined approaches to learn different components of the ontology, by using different learning algorithms such as TEXT-TO-ONTO, using association rules, formal concept analysis and clustering techniques, WEB→KB combining FOL rule learning with Bayesian learning, HASTI applying a combination of logical, linguistic based, template driven and heuristic methods and (Termier, et al., 2001) combining statistics and semantics for word and document clustering.

4. Ontology learning systems and tools

This section provides an overview of the existing implementations of ontology learning methodologies, focusing on the ontology population and ontology enrichment tools. In the context of the SOA4ALL project, it is important to be aware of the features and characteristics of current ontology tools. This information provides insight about the advantages and disadvantages of the different learning approaches and their implementations in some key areas including, level of automation, domain dependency and quality assurance. Based on this comparison, conclusions about the requirements on the ontology learning tools developed in WP3 can be made.

The following sections provide a short description of each tool, followed by a comparative analysis. The tools are organized in an alphabetical order.

4.1 Tools for ontology population

Most of the currently existing systems for ontology population are based on using an extraction toolkit, which facilitates the identification of instances of concepts, which are then assimilated into the ontology. Ontology population systems are closely related to ontology-based information extraction systems, since all ontology population systems include mechanisms that try to characterize pieces of a corpus with concepts from an ontology. In a similar way, every ontology-based information extraction system can be viewed as an ontology population system because it can easily be extended to include extracted instances into the ontology.

This section describes the most common approaches in the area of ontology learning and their implementations. Each tool is described in terms of its most important features, followed by a comparative analysis of all tools.

Adaptiva. The Adaptiva system (Brewster et al., 2002) is an ontology based user-centered pattern learning system. This system can learn patterns for extracting instances of relations between terms. Adaptiva is based on a bootstrapping approach where by starting with an initial ontology, containing at least one lexicalisation for each concept and relations between concepts, the system finds example sentences from a corpus that represent instances of relations between two concept instances. The extracted examples are validated by a human expert. After this, by employing the Amilcare system, the newly extracted examples are turned into a training corpus, where patterns for extracting relation instances are automatically learned. This process is repeated until stopped by the ontology expert.

The Adaptiva aims to extract relation instances. However, it is not described if the system is also capable of extracting concept instances. Moreover, the system does not directly perform ontology population, instead the human expert evolves the ontology based on the example sentences gathered by the system. Therefore, the problems related to ontology inconsistency, such as ontology consistency maintenance and entity disambiguation, are responsibilities of the ontology expert. Finally, the system is not restricted to only one specific domain because it does not use domain-specific resources.

(Alfonseca et al., 2006). (Alfonseca et al., 2006) developed a system for relation extraction, which automatically learns extraction patterns for finding semantic relations in unrestricted text, based on statistical corpus processing. The training corpora are processed with a part-of-speech tagger and a named-entity recognizer, which annotates persons, organizations, locations, dates, relative temporal expressions and numbers. After the initial preprocessing, the pairs of recognized named entities, which fulfill specific predefined requirements, are considered as relation instances and the context, in which the two entities appears in, is noted. Finally, the extracted patterns are filtered by additional manually-written constraints, provided as input to the system.

The system does not directly perform ontology population by using the extracted relation instances. In addition, the applicability of this system in arbitrary domains is limited because of the named entity recognition engine and the domain specific constraints for filtering the learned patterns.

Artequakt. Artequakt (Kim et al., 2002, Alani et al., 2003a; Alani et al., 2003b; Alani et al., 2003c) is a system for the automatic extraction of artifacts from the Web. It populates a knowledge base and uses this knowledge base to generate personalized biographies. Artequakt, like many other systems, makes use of a publicly available information extraction toolkit (GATE – Cunningham et al., 2002a; Cunningham et al., 2002b) to perform named entity recognition, syntactic and semantic analysis on documents, retrieved by querying common web searching engines (like Yahoo and Google). Once the instances are identified, the system uses both a domain specific ontology and a generic one (WordNet – Fellbaum, 1998) in order to extract binary relations between two instances (such as <person, date-of-birth, date>). These relations are represented in the form of triplets formulated in XML, which also constitute the output of the extraction toolkit. Unfortunately, it is not clearly described how ontology consistency is maintained. Moreover, the system does not involve machine learning and is not easily portable to new domains.

KnowItAll. The KnowItAll system is an unsupervised, domain-independent system that extracts information from the Web (Etzioni et al., 2004; Etzioni et al., 2005). During the first processing phase, the simple system uses eight domain-independent lexico-syntactic patterns (inspired by Hearst patterns – Hearst, 1992) to extract possible instances of specific concepts. During the second phase, the validity of the extracted instances is evaluated by using an extended version of the *pointwise mutual information* (Turney, 2001) statistical measure. This measure determines the instances that are used to populate the knowledge base, by querying Google in order to get the number of hits on various queries built from each instance.

The KnowItAll system is a domain-independent system because it uses a small set of domain-independent lexico-syntactic patterns in order to extract instances. Unfortunately, the KnowItAll system does not perform any relation extraction. Also, no information is provided whether the system provides consistency and entity disambiguation actions to ensure the quality of the resulting ontology.

LEILA. LEILA – Learning to Extract Information by Linguistic Analysis – is a system that learns to extract instances of arbitrary provided binary relations from natural language corpus of documents (Suchanek et al. 2006). The system uses deep linguistic analysis, which is based on broad coverage parsing grammars. The deep linguistic analysis parses the input documents and extracts patterns that can identify instances that are related to the provided input binary relation. LEILA uses an unsupervised approach; however, a classification function is required, which can classify a specific parsed segment as being a positive, a negative, a possible or not an example of the target binary relation. After the possible extraction patterns and their relevant examples have been collected, the system uses statistical techniques, including adaptive k-Nearest-Neighbor-classifiers and support vector machines, to learn the extraction patterns for the relation.

The LEILA system does not directly use the extracted instance to populate an ontology. In addition, the system has a domain non-specific implementation because it does not employ any domain-specific resource.

(Navigli and Velardi, 2006). Navigli and Velardi present a pattern-based method for the automatic enrichment of a core ontology with the definitions of a domain glossary (Navigli and Velardi, 2006a; Navigli and Velardi, 2006b). The application domain used in this system is the domain of cultural heritage. Based on this domain, the system populates the CIDOC CRM core ontology with terms extracted from glosses contained in the Art and Architecture Thesaurus (AAT). This extraction is done by using manually developed extraction patterns. During an initialization step, the system performs part of speech tagging and then named-entity recognition. This is realized by using manually created regular expressions to locate terms that can be annotated with concepts from the CIDOC CRM core ontology. During a second processing step, the system tries to determine the concept property and range of the property for each annotated text portion, in order to populate the ontology, with the help of manually developed constraining rules.

Because of its methodology and amount of manually created information required as an input, the system is domain specific. Therefore, for every new domain, new extraction patterns must be developed manually to port the system into a new domain. Ontology consistency maintenance is provided by the use of constraints imposed by the manually engineered extraction patterns.

SOBA. The SOBA system is an information extraction system (Buitelaar et al., 2006b). It is developed within the SmartWeb project and provides functionalities for automatically populating a knowledge base by using information extracted from soccer match reports found on the Web. SOBA uses a standard rule-based information extraction system, which is an enhanced version of SProUT (Drozdzynski et al., 2004), in order to extract named entities related to soccer matches and soccer-specific events. For example, the extracted named entities can be player activities, match events and match results. After the initial extraction step, the extracted events are converted into semantic structures based on special mapping declarative rules. This conversion considers also information that already may be present in the ontology, in order to reuse existing instances. In this way, the problem of entity disambiguation is solved for the resulting ontology.

The SOBA is domain specific because it utilizes manually developed, domain-specific rules for extracting information. Moreover, since the extracted information is converted into ontological semantic structures by using declarative mapping rules, these rules are most probably dependent on a specific ontology organization.

WEB→KB. WEB→KB aims to perform ontology population by browsing the Web, starting with a list of URLs. In addition to the list of URLs, the system has as input an initial ontology, containing the concepts and relations, whose instances must be extracted, and also training examples, which describe instances of these concepts and relations. By using both a statistical bag-of-words approach and a symbolic approach, the FOIL algorithm, (Quinlan, 1990) the system learns classifiers that can detect instances and relations between instances. In this way, based entirely on machine learning approaches, instances and relations are identified to populate the initially provided ontology. In contrast to most other ontology learning system, WEB→KB considers whole web pages as instances. For example, a person is represented by the “starting” web page of the set of his/her home pages. In this way, instead of employing information extraction to locate instances inside web pages, WEB→KB uses a document classification system to identify and classify as instances whole pages from web sites. In addition, for retrieving some specific information, the system uses simple information extraction, based again on machine learning. Finally, instances of relations among instances are retrieved by examining hyperlink paths that can connect two instances.

The usability of the system in different domains is partially controversial. The concept instance extraction engine is fairly generic and easy to port to new domains. On the other hand, the relation instance extraction engine uses some domain specific templates. The system does not provide any measures for securing the quality of the resulting ontology in terms of redundancy removal or existence of contradicting information.

4.2 Comparison of approaches for ontology population

Until now the main approaches and practical systems for ontology population have been presented. Each system was presented in terms of its methodology and comparison dimensions. Table 1 provides an overview of the systems based on a summary of their properties. This table includes some of the comparison dimensions as well as some general system properties, such as the leaning approach. This section provides a comparison of the described approaches for ontology population with the aim to identify features and properties important for the SOA4ALL project.

First, the system can be compared based on the elements extracted, followed by an analysis of the required input information. Some of the described systems can populate an ontology with instances of both concepts and relations (such as Artequakt, WEB→KB, SOBA, Navigli and Velardi 2006), while others concentrate only on relation instances extraction (such as Adaptiva, LEILA, Alfonseca et al., 2006). In this sense, some of the systems are capable of performing more tasks related to ontology population than others. For example, the KnowItAll system processes only concept instances. The population process which should be proposed in WP3 of SOA4ALL should be comparable with the state of the art and, therefore, be able to extract both concept and relation instances in order to populate the ontology. Therefore, the recommendation here would be to consider the characteristic of the Artequakt, WEB→KB, SOBA, and Navigli&Velardi systems for the prototypes in WP3.

Considering the input information required for each of the systems, is it better to have as less manually provided initial information and as fewer initial requirements as possible. In this way, the systems can be classified based on the resources or background knowledge requirements. Some systems do not try to learn terms and synonyms but rather uses publicly available processing resources for this task, like the Artequakt, SOBA and (Alfonseca et al., 2006). Other systems include a term/synonym extraction engine on their own, but require extraction patterns to be provided by the user (KnowItAll, Navigli and Velardi, 2006). These patterns have to be manually determined, most of the time. WEB→KB, Adaptiva and LEILA include an adaptable term/synonym extraction engine, which can be taught either with the help of concept/relation lexicalisations (Adaptiva) or through concept/relation instance examples (WEB→KB, LEILA). In the context of the SOA4ALL project, it should be considered the developed approach will most probably have semi-structured data as input. Therefore, the most suitable approach in this case would be to provide some initial input patterns, which exploit the structure of the input documents.

In addition to a classification based on the extracted elements and the input information, systems can also be compared based on the learning approach. Machine learning is used in the majority of the systems and only three systems (Artequakt, SOBA, Navigli and Velardi 2006) do not employ some form of learning. Therefore, the system that do not employ machine learning either use an external, publicly available term/synonym extraction engine or require manually constructed patterns as input. The LEILA system also uses linguistic knowledge, but employs an additional filtering based on statistical approaches. The systems, which use machine learning, either use statistical methods to identify terms, or perform automated pattern extraction. This is the also probably the approach, which will be used in SOA4ALL. In this way, the term/synonym extraction engine can make use of both linguistic information and machine learning for identifying concept instances, while automated pattern extraction can be employed for the task of relation instance extraction.

Considering other comparison criteria, the degree of automation, the population process can be fully automated, without requiring interaction with a domain/ontology expert. If domain/ontology experts are needed, the manual effort required should be reduced to a minimum. Therefore, the research work done in SOA4ALL will target the development of an automatic approach, even if the necessity of some manually provided input or supervision cannot be excluded completely.

Considering the consistency of the ontology and the prevention of redundant ontology information, most of the described systems do not address these issues at all. Only two systems (Artequakt and SOBA) take some steps towards tackling these problems. The Artequakt system follows a limited approach, by applying two manually written heuristics in order to merge instances that refer to the same real object or event. The SOBA system on performs simple checks during the instance creation directly before the instances populate the ontology, in order to re-use instances that refer to the same real object or event instead of creating new ones. In WP3, it should first be determined, how crucial it is that the learned ontology is consistent and without any redundancies. Based on this, measures for ensuring the quality of the ontology can be directly implemented or subsequent methods for improving the initially leaned ontology can be developed.

System Name	Input Information	Learned Elements	Learning Approach	Automation	Result	Domain Portability	Ontology Quality Measures
Adaptiva	*Domain ontology *Lexicalisations of all ontology concepts *Unstructured Corpus	*Relation Instances	*Pattern extraction	*Manual	*Relation instances extraction patterns	*Domain independent	*No
(Alfonseca et al., 2006)	*Linguistic knowledge *Relation constraints *Seed list of relation participating entities *Unstructured Corpus	*Instances of arbitrary binary relations	*Statistical	*Automatic	*Relation instances extraction patterns	*Domain specific	*No
Artequakt	*Named entity recognition & classification engine *Syntactic/semantic analyser *Domain ontology *Unstructured Corpus	*Concept Instances *Relation Instances	*No learning	*Automatic	*Populated Ontology	*Limited	*Limited
LEILA	*Linguistic knowledge *Instances examples for ontology concepts-relations *Unstructured Corpus	*Instances of arbitrary binary relations	*Statistical (kNNs,SVMs)	*Automatic	*Instances pairs	*Domain independent	*No
KnowItAll	*Part-of-speech Tagger *Regular expressions for shallow syntactic parsing *Domain independent lexico-syntactic patterns *Unstructured Corpus	*Concept Instances *Concepts	*Pattern extraction *Wrapper induction	*Automatic	*List of concept instances	*Domain independent	*No
SOBA	*Named entity recognition & classification engine *Information extraction engine *Domain ontology *Rules for mapping extracted information into *Unstructured Corpus	*Concept Instances *Relation Instances	*No learning	*Automatic	*Populated Ontology	*Domain Specific	*Yes
(Navigli & Velardi, 2006)	*Manually engineered extraction patterns *Domain ontology *Resources (Wordnet) *Structured Corpus	*Concept Instances *Relation Instances	*No learning	*Automatic	*Populated Ontology	*Domain Specific	*No
WEB→KB	*Domain ontology *Instances examples for ontology concepts-relations *Unstructured Corpus	*Concept Instances *Relation Instances	*Statistical *Symbolic	*Automatic	*Populated Ontology	*Fairly portable	*No

Table 1: *Comparison of approaches for ontology population*

4.3 Tools for ontology enrichment

Similarly to the tools for ontology population, in the context of the SOA4LL project, it is important to be aware of the existing implementations for ontology enrichment. The features and characteristics of current ontology tools provide insight about the advantages and disadvantages of the different learning approaches and their implementations. In this way, conclusions about the difficulties in the implementations of ontology enrichment approaches can be made and the minimum set of features of the prototypes developed in WP3 can be determined. Each tool for ontology enrichment is presented by providing a short overview of the main features, followed by a comparative analysis of all tools in the next section.

ABRAXAS. The Abraxas approach is founded on viewing ontology learning as a process involving three resources, each of which must remain in equilibrium (Iria et al., 2006):

1. **The corpus of documents**. Each document object has a structure or a set of characteristics reflecting the relationships between them in such a way that the ontological knowledge in a document is expressed using lexico-syntactic patterns. These patterns are used in such as way that the ontological relationship is automatically obtainable.
2. **The set of learning patterns**, lexico-syntactic patterns.
3. **The ontology,** described as a set of triples.

Abraxas is an event-driven system, which starts in a state of equilibrium. When this state of equilibrium is changed by a new event, such as adding a new document to the corpus or a new triple to the ontology, it triggers a new learning cycle. During this cycle the system will try and return to a new state of equilibrium, filling in the knowledge gap between corpus and ontology by using existing patterns and/or inducing new ones. The events, which influence the state of the system, are triggered by external actions specified by the user. A scheduler decides on the best action to take, whenever the system has not reached a state of equilibrium and there is no input of an external action.

The Abraxas system performs relation extraction by using automatic lexico-syntactic pattern acquisition. However, the types of the extracted relations are not described (hierarchical or non-hierarchical) and it is not clear whether the system is able to also learn concepts. Finally, there is not information available regarding the handing of consistency and redundancy problems.

ASIUM. The ASIUM system learns terms, synonyms, concepts and hierarchical relations from unrestricted text corpora, through the use of syntactic analysis (Faure et al., 1998; Faure and Poibeau, 2000). ASIUM employs a general purpose parsing grammar, which is able to detect the subject and object of a verb in prepositional phrases. Using the results of syntactic parsing, the system identifies patterns known as verb frames. Verb frames are patterns of the form "<verb> <(preposition|syntactic role): concept+>+" (Faure and Poibeau, 2000). The system assumes that nouns occurring in at least two verb frames are possible concepts. Therefore, ASIUM gathers lists of nouns that appear in at least two different verb frames. An important feature of the system is the use of similarity metric for noun lists, based on the number of common nouns in two lists. Using this metric, the system performs a supervised bottom-up breadth-first conceptual clustering in order to construct a concept hierarchy. A domain expert verifies each single step of the clustering procedure, in order to verify and name clusters.

ASIUM uses hierarchical clustering, to learn concept hierarchies. However, the approach also requires the supervision of each clustering step by a domain expert. Therefore, the domain expert effectively performs the tasks of consistency maintenance and redundancy elimination manually. Finally, the portability of the system in new thematic domains seems to only be limited by the linguistic grammar employed for the syntactic analysis.

HASTI. HASTI is an ontology learning system which extracts terms, synonyms, concepts, hierarchical and non-hierarchical relations (Shamsfard, 2003; Shamsfard and Barforoush, 2002; Shamsfard and Barforoush, 2004). The system also is capable of extracting simple axioms from unstructured textual corpus, with the help of lexico-syntactic patterns. A small domain-independent ontology serves as an input for the system. Based on this ontology, the system is able to convert sentences into ontological fragments. After the extracted concepts and relations are added to the ontology, clustering is used in order to identify and remove concept redundancy. Relation redundancy is solved with the help of heuristics. In the same way, simple heuristics are used in HASTI to handle the problem of inconsistency resolution.

The HASTI system can function in both semi-automatic and automatic modes. However, it must be pointed out, that the system does not attempt to extract knowledge specific to a given domain of interest, but rather translates all input corpus into an ontology. Rule extraction is performed in a similar manner, by creating rules from statements expressed in natural language.

(Specia and Motta, 2006). Specia and Motta describe a system for extracting instances of relations from unstructured corpora (Specia and Motta, 2006). The system uses parsing techniques and extracts triples that represent verbal relations between two entities/terms in the form <noun-phrase, verb-phrase-noun-phrase>. These entities/terms can be provided as an input to the system, or be automatically recognized through an embedded named-entity recognition system based on GATE components. After this initial step, the system employs various metrics for filtering the list of extracted triples. These metrics include similarity measures between entities and concepts in the ontology and use Wordnet synonyms for the involved verbs.

SYNDIKATE. SYNDIKATE is a system for automatically acquiring knowledge from real-world texts. The system aims to transfer the content of real-world text to formal representation structures, which constitute a corresponding text knowledge base (Hahn and Marko, 2002). SYNDIKATE requires a considerable about of input information, including a fully lexicalised dependency grammar, at the linguistic level, and a populated ontology of a considerable size, at the level of domain knowledge. Based on this input information, the system employs syntactic and semantic analysis, in addition to reasoning, in order to detect and semantically interpret terms that are unknown to the system. Every time an unknown term is detected, the contextual syntactic and semantic information is used in order to associate the unknown term with known concepts. After this step is complete, other concepts are used to “explain” the unknown term, by using reasoning. Finally, if a consistent explanation is found for the term, the term is added as a lexicalisation of an existing concept. Unfortunately, the system does not employ any form of learning because the knowledge acquisition is performed through the use of linguistic information and reasoning with the help of significant amounts of background knowledge provided in the form of input information.

TEXT-TO-ONTO. TEXT-TO-ONTO is an environment for ontology learning, whose goal is to help a domain expert in the construction of a domain specific ontology (Maedche and Staab, 2001). The system provides graphical user interfaces, which enables the domain expert to view various aspects of an ontology and to modify it. The system also employs simple linguistic analysis, in order to automate the extraction of terms and their categorization as lexicalisations of existing concepts. TEXT-TO-ONTO can function only by involving a domain expert, who is responsible to review each of the proposed instances and select the ones that should populate the ontology. When the domain expert identifies a term, which is not associated with an existing concept, the domain expert should consider the creation of a new concept, which can be associated with the unexplained term. The system is also capable of extracting hierarchical and non-hierarchical relations. The extraction of hierarchical relations is based on hierarchical clustering, the results of which must also be reviewed by the domain expert. The extraction of non-hierarchical relations, on the other hand, is based on linguistic analysis, which again must be approved by the domain expert. Finally, the domain expert can prune or manually refine the ontology by using the provided graphical user interfaces.

4.4 Comparison of approaches for ontology enrichment

The previous section provided a short overview of the existing approaches and tools in the area of ontology enrichment. In a similar way to the comparison of ontology population approaches, Table 2 presents an overview of the discussed ontology enrichment systems and their main features. The information presented in each of the columns shows the comparison dimensions, while rows contain the presented systems. The goal of this section is to provide a comparison of the approaches for ontology enrichment and identify features and properties, which can be used for achieving the objective of WP3 in SOA4ALL.

First, the system can be compared based on the elements extracted, followed by an analysis of the required input information. In terms of the elements extracted by the systems, some systems provide more thorough results, in the sense that they perform more subtasks from the “layer cake” of ontology learning aspects. ASIUM, HASTI, TEXT-TO-ONTO perform identification of new concepts, relations and in some cases even rules. On the other hand, systems like SYNDIKATE, ABRAXAS and (Specia and Motta, 2006) concentrate either on concept or relation identification. The approaches developed in SOA4ALL for ontology enrichment should correspond to the current state of the art and be as comprehensive as possible, by incorporating the needed procedures to extract concepts, hierarchical and non-hierarchical relations and rules. Therefore, the ASIUM, HASTI, TEXT-TO-ONTO system should be studied more thoroughly and features relevant for the prototypes implemented in WP3 should be determined.

Considering the input information required for each of the systems, is it better to have as less manually provided initial information and as fewer initial requirements as possible. Therefore, a desirable property of an ontology enrichment system is to have as fewer initial requirements as possible. Almost all of the here described systems rely on linguistic analysis to exploit syntactic relations in order to identify new concepts, relations or even rules. In addition to linguistic knowledge, only a few systems require additional background knowledge (SYNDIKATE, ABRAXAS). Therefore, considering ontology population, the implementations done in SOA4ALL should involve some linguistic analysis but almost no additional input information.

In addition to a classification based on the extracted elements and the input information, systems can also be compared based on the learning approach. Most of the systems use machine learning, especially in the form of clustering or in the form of lexico-syntactic pattern acquisition. Based on this observation, the prototypes developed in WP3 should definitely involve machine learning.

The degree of automation shows that the enrichment process cannot be fully automated. Most systems require the involvement of an ontology expert, except for systems that require significant background knowledge and are limited to the performed knowledge acquisition (SYNDIKATE, ABRAXAS, Specia and Motta, 2006). SYNDIKATE requires an almost complete ontology, which can be extended with new concepts originated. Limited information is available about ABRAXAS, which is reported as being able to begin even with an empty ontology. Similarly, (Specia and Motta, 2006) concentrates mainly on presenting an approach for relation identification without being a complete ontology enrichment system. The degree of automation required by WP3 is not clear yet, since the involvement of a domain expert is usually a tradeoff to the quality of the resulting ontology.

System Name	Input Information	Learned Elements	Learning Approach	Automation	Result	Domain Portability	Ontology Quality Measures
ABRAXAS	*Linguistic knowledge *Domain Ontology *Lexico-syntactic patterns *Unstructured Corpus	*Relations	*Pattern extraction	*Automatic	*Ontology (set of triples)	*Domain independent	*No Info.
ASIUM	*Linguistic knowledge *Unstructured Corpus	*Terms *Synonyms *Concepts *Hierarchical relations	*Syntactic Analysis *Conceptual Clustering	*Semi-automatic	*Concept Hierarchy	*Domain independent	*Yes
HASTI	*Linguistic knowledge *Unstructured Corpus	*Terms *Concepts *Hierarchical relations *Non-hierarchical relations *Rules	*Linguistic Analysis *Semantic Analysis *Logical reasoning *Templates	*Both automatic and semi-automatic modes	*Ontology *Rules	*Domain independent	*Yes
(Specia and Motta, 2006)	*Domain Ontology *Linguistic knowledge (swallow parser) *Resources (Wordnet) *Unstructured Corpus	*Relations	*Pattern extraction	*Automatic	*Relations	*Domain independent	*No
SYNDIKATE	*Linguistic knowledge (lexicalised dependency grammar) *Generic and domain specific lexicons *Populated Initial Ontology *Unstructured Corpus	*Terms *Concepts	*Syntactic analysis *Semantic analysis (through reasoning)	*Automatic	*Textual knowledge base *Updated lexicon *Updated ontology	*Domain dependent	*No Info.
TEXT-TO-ONTO	*Linguistic knowledge *Unstructured Corpus *Semi-structured (XML, DTD) data *Structured (DB schema, ontology) data.	*Terms *Concepts *Hierarchical relations *Non-hierarchical relations *Rules	*Morphological analysis *N-grams for multiword term detection *Hierarchical Clustering	*Semi-automatic	*Ontology	*Domain independent	*No Info.

Table 2: *Comparison of approaches for ontology enrichment*

5. Tagging approaches

Tagging-based systems enable users to categorize web resources by means of tags (freely chosen keywords), in order to re-finding these resources later. Tagging is implicitly also a social indexing process, since users share their tags and resources, constructing a social tag index, so-called folksonomy.

At the same time, an interface model for visual information retrieval known as Tag-Cloud has been popularised. This model displays the most frequently used tags in alphabetical order.

In the context of SOA4ALL, the idea is to propose a method to model folksonomies with ontologies. The method will be composed of, on one side, a generic ontology structure that will represent any folksonomy, and, on the other side, an algorithm that integrates the information contained in the folksonomy with the generic ontology. Thus, it is necessary to develop an algorithm for obtaining an ontology that contains the tagged information from the folksonomy.

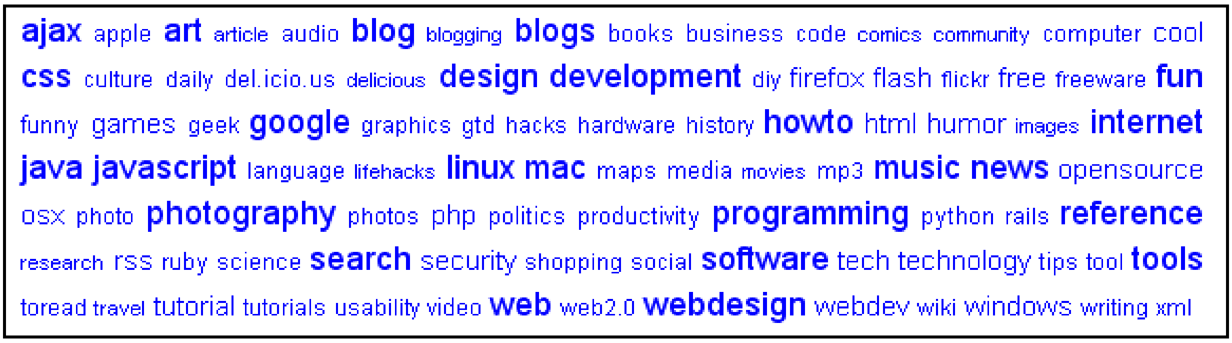
Along this section we will present the aforementioned issues (tags and tagging-based systems, folksonomies and tag-cloud), as well as the state of the art on ontologies for semantic tag. Finally, we will conclude with the advantages and disadvantages of social tagging.

5.1 Tags and Tag Cloud

A tag is a relevant metadata keyword or term associated with, or assigned to, an object such as a picture, article, or video clip, thus describing the item and enabling keyword based classification of information it is applied to. As opposed to authoritative metadata, as used for library cataloguing, tags are metadata keywords that end-users add. The process of adding tags is called tagging.

Tags are usually chosen informally and personally by the author, creator, user or consumer of the item. Here the focus is not on a library cataloguer and the author's intent; the idea of tags is to try to represent more the end-user's intent. Moreover, unlike cataloguing terms, tags are not usually part of some formally defined classification scheme.

A tag cloud (or weighted list in visual design) can be used as a visual depiction of content tags used on a website (Figure 2). Often, more frequently used tags are depicted in a larger font or otherwise emphasised (bold, colour, etc), while the displayed order can also be alphabetical. Thus, both finding a tag in an alphabetic list and by popularity is possible. In a Tag-Cloud, when a user clicks on tag obtains an ordered list of tag-described resources, as well as a list of others related tags. Whereas querying requires to user to formulate previously his information needs, visual browsing allows the user to recognize his information needs scanning the interface. Visual browsing is similar to hypertext browsing in the way that both of which allow the user to search by browsing. However there is a difference, that is, visual interfaces provide a global view of tags or resources collection, a contextual view.



ajax apple art article audio blog blogging blogs books business code comics community computer cool
css culture daily del.icio.us delicious design development diy firefox flash flickr free freeware fun
funny games geek google graphics gtd hacks hardware history howto html humor images internet
java javascript language lifehacks linux mac maps media movies mp3 music news opensource
osx photo photography photos php politics productivity programming python rails reference
research rss ruby science search security shopping social software tech technology tips tool tools
toread travel tutorial tutorials usability video web web2.0 webdesign webdev wiki windows writing xml

Figure 2: *Traditional Tag-Cloud.*

Tag-Cloud is a simple and widely used visual interface model, but with some restrictions that limit its utility as visual information retrieval interface. It is due to:

- The method to select the tag set to display is based exclusively on the use frequency, which inevitably entails that displayed tags have a high semantic density. In terms of discrimination value, the most frequently-used terms are the worst discriminators (Salton, G.; Wong, A. and Yang, C.S, 1975). As indicate (Begelman, G.; Keller, P. and Smadja, F., 2006), very few different topics, with all their related tags, tend to dominate the whole cloud. (Xu, Z. et al., 2006) suggest also the need of research on tag selection methods in order to improve Tag-Clouds.
- Alphabetical arrangements of displayed tags neither facilitate visual scanning nor enable infer semantic relation between tags. Probably, similarity-based layout would improve Tag-Cloud browsing. Although a folksonomy (section 5.2) is commonly defined as a flat space of keywords without previously defined semantic relationships between tags, different studies (Mika, P., 2005) (Brooks, C.H. and Montanez, N., 2006) (Shaw, B., 2005) demonstrate that associative and hierarchical relationships of similarity between tags can be inferred from tag co-occurrence analysis (Salton, G.; McGill, M.J., 1983).

The first widely known use of tag clouds was on the photo-sharing website Flickr. That implementation was based on Jim Flanagan's Search Referral Zeitgeist, a visualization of web site referrers. Tag clouds have also been popularised by Delicious and Technorati, among others and have also triggered a variety of visualisation approaches to accessing resources.

The first published appearance of a tag cloud can be attributed to the "subconscious files" in Douglas Coupland's Microserfs (1995). A tag cloud can be: a personal one comprised only of personal keywords; a local one to display the most common tags by a given community of users (e.g. based on domain interest, language, etc.); or even a global one, allowing access to all resources that are available through the service.

5.2 Folksonomies

A folksonomy is an Internet-based information retrieval methodology consisting of collaboratively generated, openended keywords, e.g. tags or labels, that categorise any content, such as Web-pages, online photographs and Web-links. A folksonomy is most notably distinguished from a taxonomy in that the authors of the tagging system are often the main users, and sometimes originators of the content, to which the tags are applied. This differs from the classical library setting where professional librarians are in charge of cataloguing, i.e. adding metadata and keywords, but are not the main users of the cataloguing system, nor originators of the content.

Folksonomies arise in Web-based communities where special provisions are made at the site level for creating and using tags, i.e. the Website or service supports user-generated metadata. Good examples are sites like Delicious or Flickr. Generally, these users can tag two types of content; user generated or originated content, such as photographs, blog postings, etc., or they collaboratively tag existing content, such as Websites, books, scientific and scholarly literature. Folksonomies develop from the tags that these communities use.

Folksonomy should be distinguished from folk taxonomy, a cultural practice that has been widely documented in anthropological work. Folk taxonomies are culturally supplied, intergenerationally transmitted, and relatively stable classification systems that people in a given culture use to make sense of the entire world around them (not just the Internet) (Berlin, 1992).

The term folksonomy is a portmanteau that specifically refers to the tagging systems created within Internet communities. A combination of the words folk (or folks) and taxonomy, while "Folk" is from the Old English folc, meaning people and "Taxonomy" being the science or technique of classification (Wikipedia, 2007).

Folksonomy allows anyone to access to any web resource that was previously tagged, based on two main paradigms of information access: Information Filtering and Information Retrieval.

In IF user plays a passive role, expecting that system *pushes* or sends toward him information of interest according to some previously defined profile. Social bookmarking tools allow a simple IF access model, where user can subscribe to a set of specific tags via RSS/Atom syndication, and thus be alerted when a new resource will be indexed with this set.

On the other hand, in IR user seeks actively information, *pulling* at it, by means of querying or browsing. In tag querying, user enters one or more tags in the search box to obtain an ordered list of resources which were in relation with these tags. When a user is scanning this list, the system also provide a list of related tags (i.e. tags with a high degree of co-occurrence with the original tag), allowing hypertext browsing.

5.3 Overview of Tag Ontologies

A lot of scholarly work has been done on the topics of folksonomies and the Semantic Web, but relatively few studies have been carried out on tagging representation at a semantic level. However, a formal representation for tagging plays important roles to reflect various experimental results on the Web. Many studies have been performed in a lot of disciplines via innovative approaches. However, without consistent structures and semantics, contributions of these studies are unable to analyze the social phenomenon relating the folksonomies. In order to operate social ecosystems on the Web, we need various technical and social analyses for folksonomies as well as formal representation for adopting the results. The semantics of tagging data is primarily about an agreement on the meaning among people or a community in the social space. A common semantics provides a way to share tag representation among services. We now provide an overview of a number of existing efforts that had the common aim of representing the concepts and operations of tags and tagging.

5.3.1 Tag Ontology

Using the tripartite *Tagging(User,Resource,Tag)* model Newman et al. (Richard Newman, Danny Ayers, and Seth Russell, 2005) defined an ontology of tags and tagging, simply called the Tag Ontology, that describes the relationship between an agent, an arbitrary resource, and one or more tags. Thus, in his ontology, the three core concepts Taggers, Tagging, and Tags are used to represent the tagging activity.

Notably, in this ontology tags are represented as instances of the `tags:Tag` class which is assigned custom labels, i.e. the string representing the tag as seen by the user. Being instances of a class means that they are assigned a URI. URIs are a key feature of the Semantic Web, since, contrary to simple literals, they can be used as subject of triples, while literals can be only used as objects. This way, tags - identified by URIs - can be linked together and people can semantically represent connection and similarities between tags. For this purpose the ontology introduces a `tags:related` property. Yet, this relation does not have much semantics, since it does not define the nature of the relation, e.g. if this a linguistic variation or because it identifies a similar topic. Another limitation is that the ontology does not define any cardinality constraint on the number of labels a Tag can have. This can raise problems since it allows a Tag instance to have two completely disjoint labels (i.e. a Tag instance with labels “RDF” and “Paris”), which makes no sense from a tagging point of view.

Still, this ontology reuses pre-defined Semantic Web vocabularies, making it compliant with existing standards. SKOS properties are used to model relations between tags and the Tag class itself inherits from `skos:Concept`. DublinCore is used to represent the date of a tagging action, with subproperties of dc:date. Finally, the ontology relies on FOAF to identify the tagger of a tagging action thanks to `foaf:Person`.

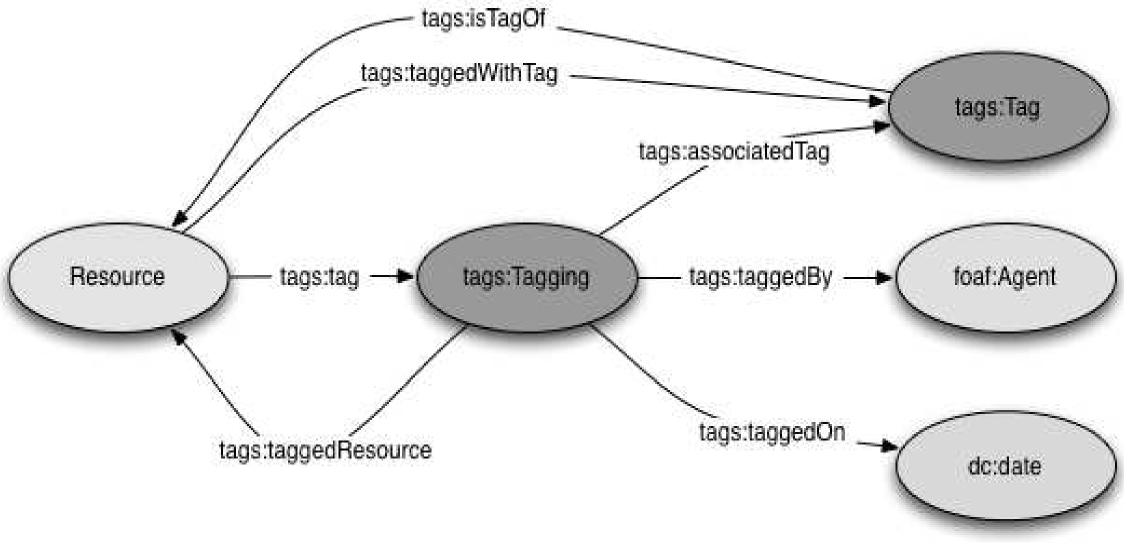


***Figure 3:** Tagging activity using Tag Ontology*

5.3.2 SCOT

The Social Semantic Cloud of Tags ontology aims to describe the structure and the semantics of tagging data and to offer social interoperability of the data among heterogeneous sources.

Both Tagcloud and Tag class in SCOT play a role to be able to represent social and semantic context of tagging, since both classes include users, tags, and resources and additional information to clarify tags’ semantics. `scot:TagCloud` has properties that describe a certain user, tag spaces, number of tags, posts and co-occurrences and their frequencies, as well as updated information. The property `scot:contains` links `scot:TagCloud` to a set of `scot:Tag` instances. `scot:Tag`, as a subclass of `tags:Tag` from the Tag Ontology, describes a tag that is aggregated from individual tagging activities. This class has several properties such as `scot:spelling_variant`, `scot:synonym` to solve tag ambiguities. We called these properties “linguistic property” since they focus on representing the meaning of the relationship between tags from a linguistic point of vie. In addition, this class has also properties to describe occurrence of a tag (i.e. `scot:frequency`). A tag itself has its own frequency.

The frequency is not unique, but it is an important feature to distinguish or compare with other tags. We called it a “numerical property” The properties have their own numerical values by computing.

It is important to note that SCOT uses concepts and properties of Newman’s model. As shown in Figure 3, the Tagging class represents tags themselves, the resources that are being tagged, and the users that create these tags (`tags:taggedBy`). The `scot:Tagcloud` class connects `tags:Tagging` instances via the property `scot:tagging_activity`. In SCOT, we try to define the range values of tagging properties more specifically.

For instance, `tags:taggedResource` has `sioc:Item` as a range value whereas `tags:associatedTag` has `scot:Tag` as its range. Individual tags in `tags:Tagging` are mapped to a resource with `scot:Tag` instance and then these tags are represented by a collection of tags underlying a `scot:Tagcloud`. Moreover, the property `scot:tagging_account` represents an account of users in online services. Figure 4 illustrates the SCOT ontology model with integrating Newman's model.

At an individual level, created tags are ad-hoc/informal and must be relevant to a tagger and the tagger's view. Thus, tag sharing has to support aggregated views while keeping the local context of each tagging activity. SCOT provides a tagging social structure for seamless tag sharing across heterogeneous users, applications or sources. For instance, suppose that a user uses the tag 'web' three times in three different instances of `tags:Tagging`. The instance for the tag gather each tag with the URI, the property `scot:aggregated_tag` has URIs for the each tag and the property `scot:tag_of` is linked to `sioc:Item`. Moreover, this class represents not only a tag label, but also its absolute and normalized occurrences, spelling variants, and hierarchical structures (i.e. `skos:broader` and `skos:narrower`) with SKOS.

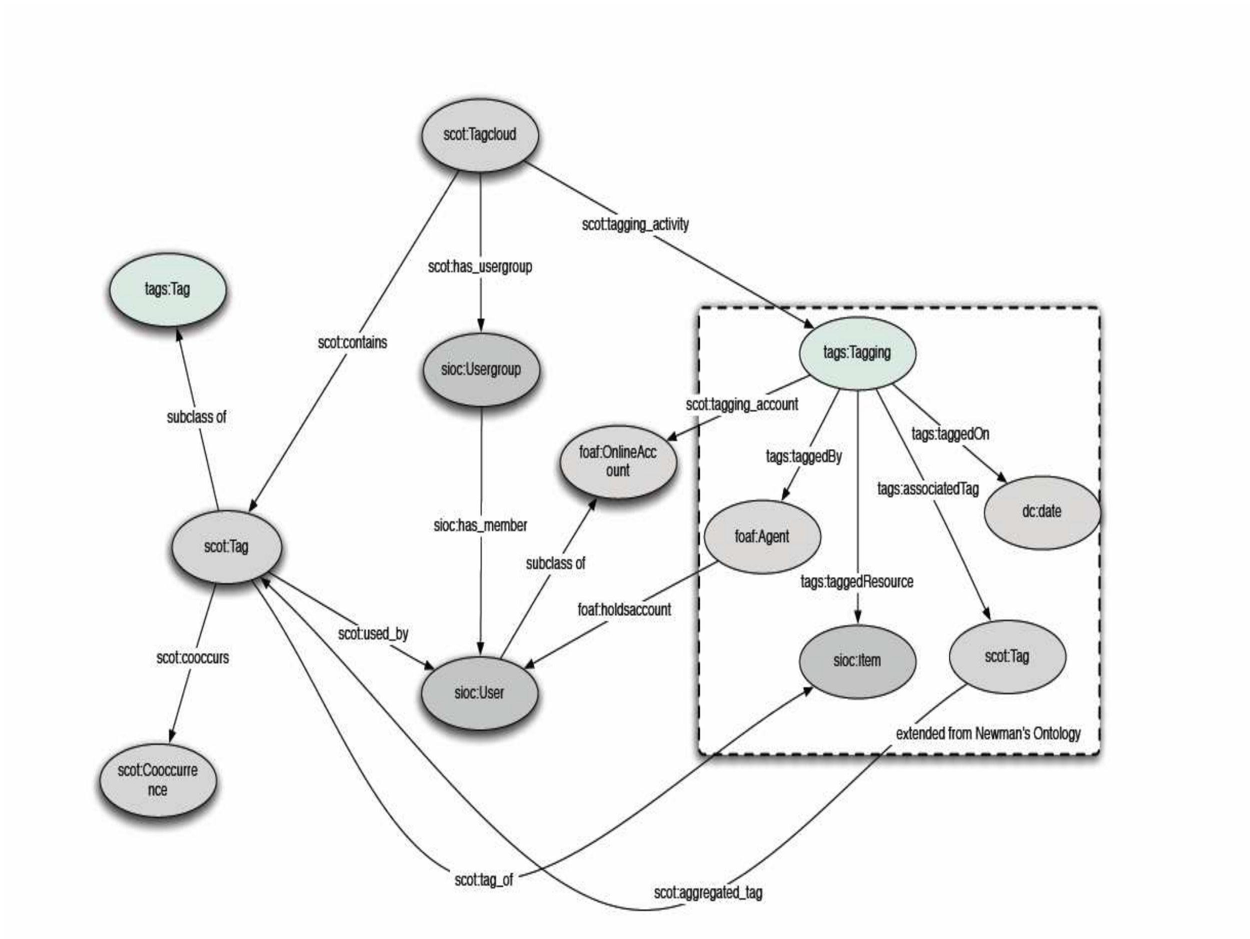


Figure 4: *Simplified folksonomy model in SCOT*

SCOT allows the exchange of semantic tag metadata for reuse in social applications and enables interoperation amongst data sources, services, or agents in a tag space. These features are a cornerstone to being able to identify, formalize, and interoperate a common conceptualization of tagging activity at a semantic level. SCOT Exporter for WordPress can extract tagging data from WordPress and then generate SCOT instance in the blog. This process is totally automatically performed with real time updates. So far, the SCOT ontology has been used in the interest web site which provides search, bookmarking and integrating tagging data among heterogenous users, sources, or applications (Kim H. L, et al., 2007).

5.3.3 MOAT

MOAT's goal is to provide a Semantic Web model to define the meaning of tags in a machine-readable way. To achieve it, MOAT defines:

- the global meanings of a tag, i.e. the list of all meanings than can be related to a tag in a complete folksonomy;
- the local meaningof a tag, i.e. the meaning of a tag in a particular tagging action.

Indeed, for instance, the tag “paris” can mean – depending on the user, the context and other factors - a city in France, a city in the USA, or even a person. Yet when someone uses it in a tagging action, it has a particular meaning, for example the french capital. Thus, MOAT extends the usual tripartite model of tagging action to the following quadripartite model: *Tagging(User,Resource,Tag,Meaning)*. Using MOAT, those meanings (both global and local) can be defined without ambiguity by the tagger. MOAT provides a machine-readable format, using a particular ontology, to allow computers to understand these meanings, relying on URIs of existing concepts from knowledge bases as DBpedia, GeoNames, or even corporate knowledge bases to define it.

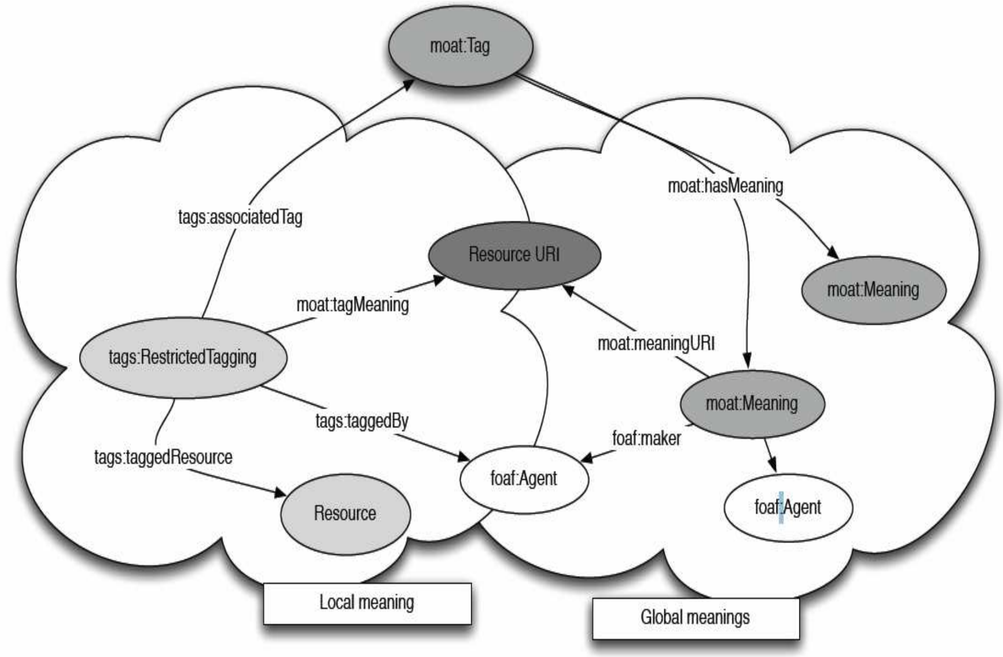


Figure 5: *Tags' local and global meaning in MOAT*

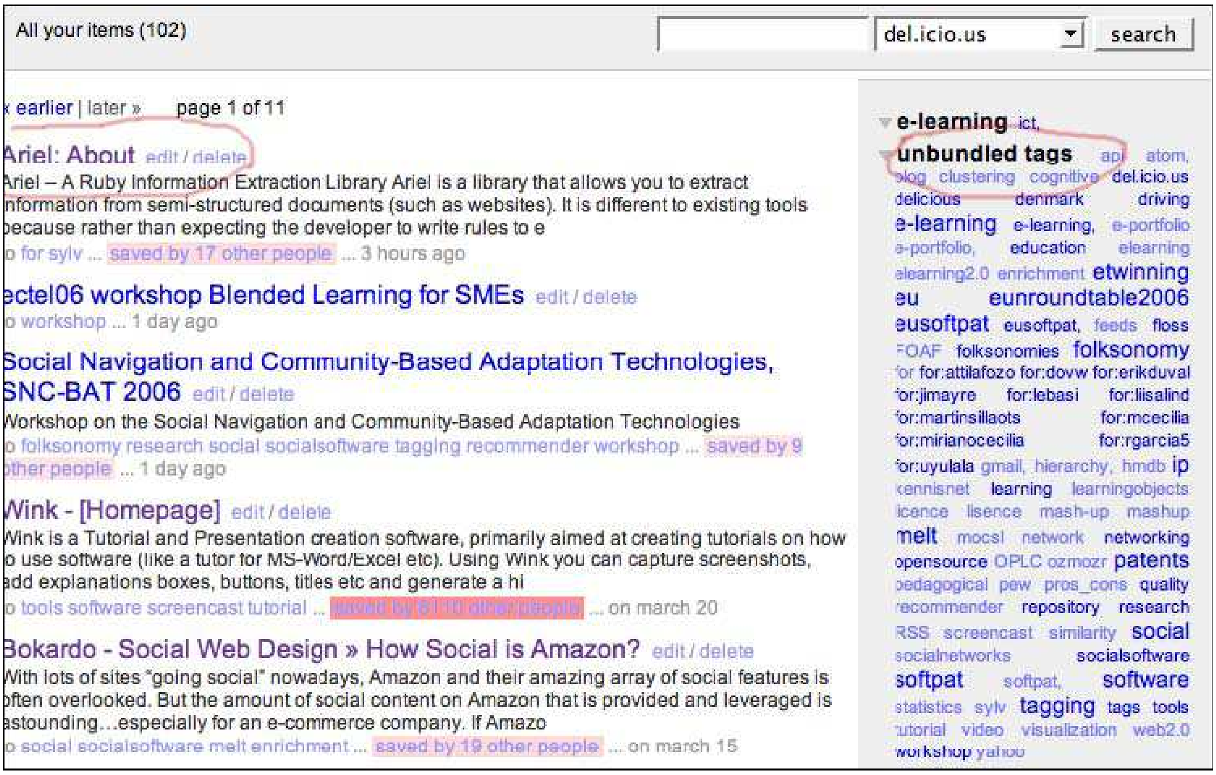
Figure 5 shows how MOAT models those meanings and reuses the Tag Ontology. MOAT introduces a Tag class as a subclass of Newman's Tag one. This subclass addresses one of the problem of the Tagging Ontology we referred to earlier, and through an OWL cardinality constraint it is only allowed _to have one unique label for a given Tag instance. Each tag is linked to one or more `moat:Meaning` instances, which represent the meaning(s) of a tag without any context. Each meaning must have one unique URI identifying it, and be linked to the agents that defined this meaning, relying on FOAF. In that way, in the folksonomy space, meanings of tags are related to the URIs of people who assigned it: *Meanings(Tag) = {(Meaning, {User})}*. To represent the context of a tag in a certain tagging action, using the quadripartite model defined before, MOAT relies on the `tags:RestrictedTagging` class from Newman's ontology, and introduces a `moat:tagMeaning` property that allows to link to the meaning of the tag in this particular context. Moreover, MOAT introduces a social aspect that lets people share their tags - and their meanings - within a community by subscribing to a MOAT server, as they could do with the Annotea annotation server. They subscribe to a tag server in which they can share and update tag meanings, and use it when tagging content. When a user tag content, the client queries the server to retrieve tag meanings and let the user choose which one is the most relevant one, regarging the context.

While it provides a model to represent those meanings, there is no automation, or Natural Language Processing to assist the user in chosing a new or existing tag when querying a post. Yet, the MOAT client for Drupal features interaction with the Sindice search engine to help users chose a new URI if no relevant meaning is found. MOAT will also, in the future, rely on social networking to give higher priority to tags used by friends within a community when suggesting meanings for a given tag.

Thanks to this framework and its model, MOAT aims to provide an easy way to bridge the gap between free-tagging and semantic indexing. While users can still benefit from the simplicity of free-tagging when annotating content, linking to URIs offers a way to solve tagging ambiguity (a single tag can be related to different URIs) and heterogeneity (various tags can be related to a single URI). Moreover, using MOAT, tagged content can be linked to URIs of reference datasets, leveraging tags and tagged content to the Linked Data web. Then, relationships defined in those datasets can be used to suggest relevant content, e.g. suggesting posts tagged "paris" from posts tagged "france" since related concepts are interlinked in DBpedia, solving the usual problem of lack of organisation in tag systems.

5.4 Tagging Functional Overview

A tagging system enables users to add keywords, or tags, to digital resources over the Internet. It also allows users to collect, store and organise these resources and retrieve them using the tags applied. This principle is similar to any cataloguing system that, for example, a library or repository uses, but it lacks a pre-defined taxonomy structure that is characteristic to cataloguing and categorising systems. What makes social tagging systems different from conventional indexing approaches is the fact that they support social interactions. Social tagging systems allow users to connect to other users, to their resources and tags that they have saved. This connection happens through relationships that form between users, their resources and tags.



All your items (102)

« earlier | later » page 1 of 11

Ariel: About edit / delete
Ariel – A Ruby Information Extraction Library Ariel is a library that allows you to extract
information from semi-structured documents (such as websites). It is different to existing tools
because rather than expecting the developer to write rules to e
o for sylv ... saved by 17 other people ... 3 hours ago

ectel06 workshop Blended Learning for SMEs edit / delete
o workshop ... 1 day ago

Social Navigation and Community-Based Adaptation Technologies, SNC-BAT 2006 edit / delete
Workshop on the Social Navigation and Community-Based Adaptation Technologies
o folksonomy research social socialsoftware tagging recommender workshop ... saved by 9 other people ... 1 day ago

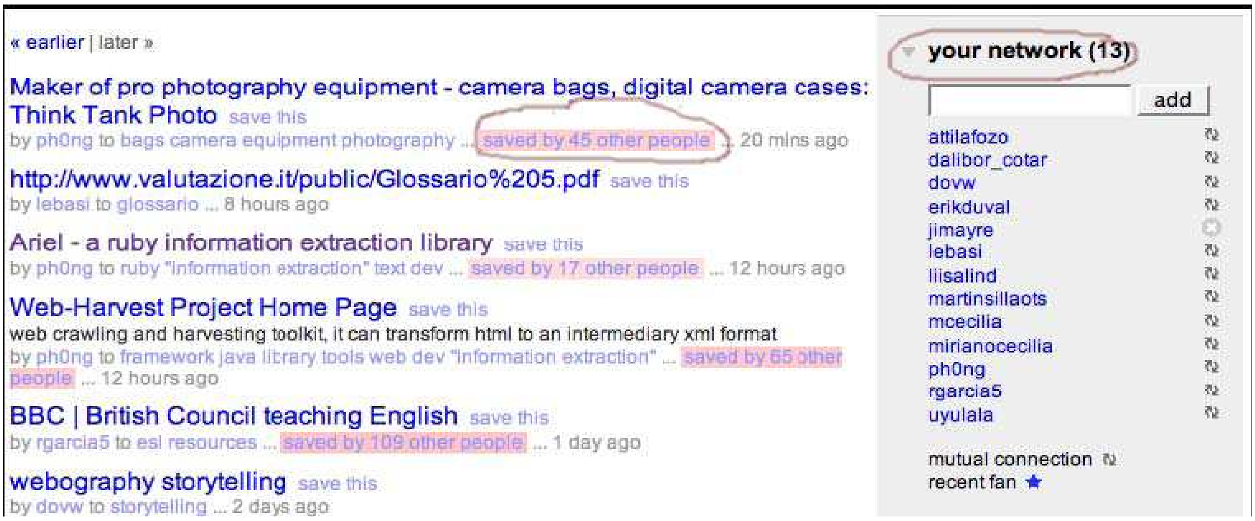
Wink - [Homepage] edit / delete
Wink is a Tutorial and Presentation creation software, primarily aimed at creating tutorials on how
o use software (like a tutor for MS-Word/Excel etc). Using Wink you can capture screenshots,
add explanations boxes, buttons, titles etc and generate a hi
o tools software screencast tutorial ... saved by 6110 other people ... on march 20

Bokardo - Social Web Design » How Social is Amazon? edit / delete
With lots of sites "going social" nowadays, Amazon and their amazing array of social features is
often overlooked. But the amount of social content on Amazon that is provided and leveraged is
astounding…especially for an e-commerce company. If Amazo
o social socialsoftware melt enrichment ... saved by 19 other people ... on march 15

e-learning ict,
unbundled tags api atom, blog clustering cognitive del.icio.us delicious denmark driving e-learning e-learning, e-portfolio e-portfolio, education elearning elearning2.0 enrichment etwinning eu eunroundtable2006 eusoftpat eusoftpat, feeds floss FOAF folksonomies folksonomy for for:attilafozo for:dovw for:erikduval for:jimayre for:lebasi for:liisalind for:martinsillaots for:mcecilia for:mirianocecilia for:rgarcia5 for:uyulala gmail, hierarchy, hmdb ip kennisnet learning learningobjects licence lisence mash-up mashup melt mocsl network networking opensource OPLC ozmozr patents pedagogical pew pros_cons quality recommender repository research RSS screencast similarity social socialnetworks socialsoftware softpat softpat, software statistics sylv tagging tags tools tutorial video visualization web2.0 workshop yahoo

Figure 6: *Users can access the system by bookmarks or tags...*

For example, in the Delicious social bookmarking service, the above mentioned relationships can be used to access resources in the system; a user can access the resources in the system through bookmarks and tags (Figure 6) and other users (as in Figure 7).



« earlier | later »

Maker of pro photography equipment - camera bags, digital camera cases: Think Tank Photo save this
by ph0ng to bags camera equipment photography ... saved by 45 other people ... 20 mins ago

http://www.valutazione.it/public/Glossario%205.pdf save this
by lebasi to glossario ... 8 hours ago

Ariel - a ruby information extraction library save this
by ph0ng to ruby "information extraction" text dev ... saved by 17 other people ... 12 hours ago

Web-Harvest Project Home Page save this
web crawling and harvesting toolkit, it can transform html to an intermediary xml format
by ph0ng to framework java library tools web dev "information extraction" ... saved by 65 other people ... 12 hours ago

BBC | British Council teaching English save this
by rgarcia5 to esl resources ... saved by 109 other people ... 1 day ago

webography storytelling save this
by dovw to storytelling ... 2 days ago

your network (13)

attilafozo
dalibor_cotar
dovw
erikduval
jimayre
lebasi
liisalind
martinsillaots
mcecilia
mirianocecilia
ph0ng
rgarcia5
uyulala

mutual connection
recent fan ★

Figure 7: *...or thorough traces left by other users.*

The first abstract model of such systems was only offered in late 2006. (Marlow et al., 2006) propose a model for the underlying structure of the tagging system. Three main components were identified; resources, tags and users, and relationships can be studied.

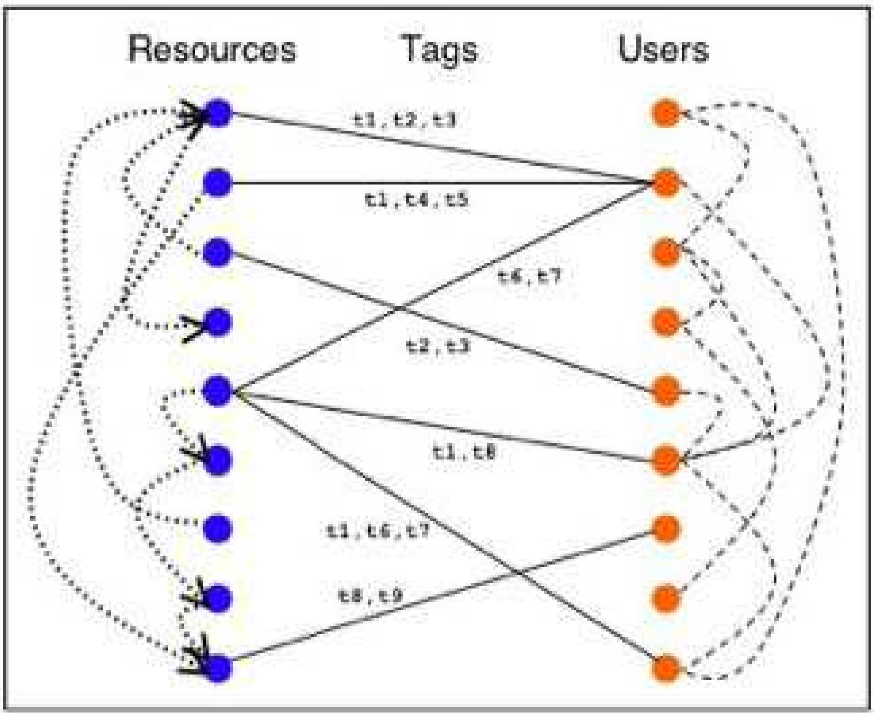


***Figure 8:** A model of tagging system (Marlow, et al., 2006).*

Figure 8 provides a conceptual model for social tagging systems. In this model, users assign tags to a specific resource; tags are represented as typed edges (i.e. links) connecting users and resources. Resources may also be connected to each other (e.g., as links between web pages) and users may be associated by a social network, or sets of affiliations (e.g., users that work for the same company). (Marlow, et al., 2006)

Through a comparison of several tagging systems, Marlow et al. (2006) shows that the socio-technical design of the system will affect the information it generates. For the purpose of designing such systems he proposes two taxonomies:

- System design and attributes. We claim that the place of a tagging system in this taxonomy will greatly affect the nature and distribution of tags, and therefore the attributes of the information collected by the system.
- User incentives. User behaviours are largely dictated by the forms of contribution allowed and the personal and social motivations for adding input to the system. The place of a tagging system in this taxonomy will affect its overall characteristics and benefits."

The process of collaborative tagging has been influenced by the advances in network technologies, where it is possible to distribute the task amongst the maximum number of individuals or computers possible.

"Just as we figured out that scanning outer space for intelligent life signals is a task that can proceed more efficiently by being distributed across many computer processors, we have begun to realize that other tasks that require human involvement can also be distributed across individuals by using the largest human network in history: the internet. This principle of distribution is at work in sociotechnical systems that allow users to collaboratively organize a shared set of resources by assigning classifiers, i.e. tags, to each item." (Meijas, 2004).

Folksonomies attempt to make a body of information increasingly easier to search, discover, and navigate over time - the ultimate challenge being to organise the Internet in general. The core idea is that, when many people are involved in the process of collaborative tagging, the community is able to advance the task faster than individual indexers, cataloguers and librarians all together.

5.5 Tagging: Advantages and Disadvantages

Social tagging potentially offers advantages in terms of:

- Personal knowledge management: As the user is able to devise terms that are meaningful to her/him, this results in more intuitive categorisation and retrieval of the content under consideration. By adopting one's own tagging vocabulary, the intention is to make a body of information increasingly easier to search, discover, and navigate over time. It is worth mentioning that this point clearly distinguishes tagging from traditional classification actions, where the roles of an author, reader and the classifier are much more distinct. In addition, tags not only contribute to personal knowledge management, but also to knowledge management for the entire community of teachers and learners.
- Serendipitous access to resources: Tags and folksonomies are useful entry points to explore the content by searching, filtering, navigating, and exploring other users' tags and tagged items. According to (Golder and Huberman, 2006), tagging is like filtering; out of all the possible items that are tagged, a filter (i.e. a tag) returns only those items identified with that tag. Depending on the implementation and query, a tagging system can, instead of providing the intersection of tags (thus, filtering), provide the union of tags; that is, all the items tagged with any of the given tags. From a user perspective, navigating a tag system is similar to conducting keyword-based searches; regardless of the implementation, users are providing salient, descriptive terms in order to retrieve a set of applicable items. Moreover, tags can offer interesting ways to browse content in terms of tag clouds or by browsing items that other people have tagged and saved. In some implementations (e.g. Furl, 'my recommendations'), the system even recommends a list items, with indications of hot or cold (related to the predicted interest) to users from other similar users. Also, because folksonomies develop in Internet-mediated social environments, users are able to discover who created a given tag, and see the other tags that this person created. In this way, folksonomy users can discover the tag sets of another user. The result, often, is an immediate and rewarding gain in the user's capacity to find related content.
- Enhanced possibilities to share content with emerging social networks: Tags can help people to share their tagged items with others. Many social tagging services allow users to create networks with other users that they have identified as interesting. Moreover, one can become a subscriber of someone else's tagged items through Web-based feeds or email notifications.

On the other hand, we can mention some of the main critics or disadvantages of social tagging:

- No standard set of keywords (also known as controlled vocabulary).
- No standard for the structure of such tags (e.g. singular vs. plural, capitalization, etc.).
- Mistagging due to spelling errors.
- Polysemy (words which have multiple related meanings - for example, a mole can be both a small mammal and a person who has infiltrated an organisation and is passing on confidential information)
- Unclear tags due to synonyms (tv and television, or Netherlands/Holland/Dutch).

- Antonym confusion (a word having a meaning opposite to that of another word: the word wet is an antonym of the word dry).
- A lack of mechanisms for users to indicate hierarchical relationships between tags (e.g. a site might be labelled as both cheese and cheddar, with no mechanism that might indicate that cheddar is a refinement or sub-class of cheese).

According to (Guy, M., & Tonkin, E., 2006), social tagging and folksonomies all but invite deliberately idiosyncratic tagging, called ‘meta noise’, which burdens users and decreases the systems’ information retrieval utility. Those who prefer top-down taxonomies/ontologies argue that an agreed set of tags enables more efficient indexing and searching of content. (Wikipedia, Guy, 2006).

6. Choosing an appropriate learning system

The main goal of WP3 is to provide methods and an implementation for the semantic annotation of Web services. This involves the determining of the relevant sources of information for the semantic annotation of services, the development of a specific methodology for the extraction of service annotations based on the input information and the refinement of the service annotations. The here presented ontology learning approaches and tools are important for the second step, the development of a methodology for service annotation. The goal is to facilitate the annotation of services by first developing an ontology (ontology leaning) based on the input information and then populating the ontology (ontology population). As a result, the semantic annotation of one service is based on the ontology instances resulting from the input documents related to this particular service. Therefore, the determining of an appropriate leaning system is crucial for achieving the tasks of WP3.

After providing an overview of the existing ontology population and enrichment approaches and performing a comparative analysis of their features, it is important to determine the characteristics and properties necessary for achieving the SOA4ALL tasks. In order to build a suitable ontology for an application, it first has to be determined what input information is available and what the requirements on the resulting ontology are. After this initial step, a methodology must be developed for using the available input information and achieving the determined goals. The implementation can use existing ontology learning systems or require the development of a new one. The objective of this section is to determine the requirements on the ontology learning component necessary in SOA4ALL by considering the input information, the resulting ontology, and the actual learning process.

6.1 The input information

The determining of the necessary input information depends on both availability and necessity of background knowledge. In environments for which no background knowledge is available, such as domains for which no base ontology is developed or languages for which no semantic lexicon is available, a new ontology has to be developed from scratch. The development of a completely new ontology has the advantage of not having to deal with integration problems. The main disadvantage of this approach is that it takes more time to learn everything from scratch and it is difficult to resolve ambiguities because of the lack of knowledge. In the case of SOA4ALL, it will not be necessary to start without any background information. It is already determined, that WSMO and WSMOLight will be used for the semantic Web service descriptions. In addition, there is already some research work done specifically for the ontology learning in the context of semantic Web service descriptions. Moreover, since the documents describing a service will be collected from the Web, the input data will be semi-structured (HTML, XML).

6.2 The resulting ontology

Depending on the different applications, different ontologies have to be developed. Ontologies may differ in their contents as well as in the activities that they support, such as logical reasoning. For instance scientific, financial or business problem solving applications usually use small, narrow, deep ontologies with specialized details coded in axioms to solve their specific problems. On the other hand, some information retrieval systems simply require wide but superficial ontologies, containing concept hierarchies with few inter-relations, and without axioms such as, WordNet or (Sowa, 2000)'s conceptual system. In the case of SOA4ALL and WP3, which consider a specific area of interest, namely, Web services small, narrow, deep ontologies with specialized details coded in axioms should be most suitable.

6.3 The learning process

As the overview of the available approaches and tools shows, although there are some fully automatic systems, they are very restricted, work under limited circumstances and have lower performance compared to semi automatic systems. Therefore, systems, which involve domain experts, give much more acceptable results because some interpretation decisions are left to the user during the learning process. Because of this, the research work in WP3 should be oriented toward developing a semi-automatic leaning process. The problems of fully automatic approaches come from the limitations of linguistic tools, which combined with the particular nature of the language, result in not so high precision values. In such cases it is crucial to involve a domain expert in order to improve the quality of the resulting ontology.

After considering whether to choose an automatic or a semi-automatic learning process, the learning approach itself also has to be determined. In learning approaches there is the choice between statistical and symbolic methods. Statistical methods are more computable, more scalable and easier to implement, while symbolic approaches are more precise, more robust and give more reasonable results. Statistical methods more generally applicable and can be used for different domains or languages. Symbolic methods, on the other hand, such as linguistic based or pattern driven methods, need more adaptations. Based on the leaning approaches and implementation described in this document, the recommendation for WP3 is to develop a hybrid approach mainly based on machine learning but also including some pattern-based recognition, in order to make use of the semi-structured nature of the input data.

Summarized, appropriate learning system for WP3 would have as input semi-structured data, employ machine learning methods and pattern-based recognition, be supervised by a domain expert and have as output a relatively small and narrow, but deep ontology. The resulting ontology can then be used for the annotation of Web service. It must be pointed out that the goal of WP3 is not to develop a complete leaning system but rather to implement some components, which perform ontology learning tasks. The recommendations for these components were determined in this deliverable.

7. Conclusion

This deliverable presents and overview of the approaches and tools available in the area of ontology learning. The choosing of the proper method for ontology learning will ensure the successful completion of the tasks envisioned in work package 3, since the annotation of services is highly dependent on the ontology, on which these annotations are based. Special emphasis was given to the subtasks of ontology population and ontology enrichment. All of the described approaches were characterized based on features that allowed the comparative presentation.

The analysis of the different advantages and disadvantages of both population as well as enrichment systems provides some important insights which can be used in the SOA4ALL project. First, it was determined that it is necessary to specify the type and source of required input information, including background knowledge and additional resources. Since, the documents describing service are collected from the Web, the input information has semi-structured nature (HTML, XML). Moreover, patterns, which recognize terms based on the structure of the documents, should also be provided as input.

Second, it was observed that most of the ontology learning processes heavily rely on linguistic preprocessing, especially syntactic analysis and exploitation of additional resources like thesaurus and semantic hierarchies like WordNet. In the context of WP3 it is suggested to employ a pattern based approach, especially for the extraction of relations between concepts, in combination with machine learning. However, even if machine learning is very commonly used for modeling various aspects of the ontology learning process, many systems still require manual intervention. Therefore, it is recommended that for WP3 a domain expert is involved, in order to improve the quality of the resulting ontology by taking final decisions on identified ontology elements. Finally, ontology population tasks seem to require less manual intervention, effectively automating a large portion of the population process. The population process which should be proposed in WP3 of SOA4ALL should be comparable with the state of the art and, therefore, be able to extract both concept and relation instances in order to populate the ontology.

The method used in the ontology learning process should take advantage of the tagging technology discussed in section 5. It should involve users associated with the input sources and allow for a transparent capture of users' knowledge. In practice, the scenario envisioned would be two-fold. On the one hand, the user independently annotates the content, resources, documents, and so on. On the other hand, the modelling mechanism stores the information in an ontology without user intervention and, moreover, dynamically (that is to say, ideally the mechanism could run in the background, while the user performs annotation).

The deliverable provides an analysis of the state of the art in ontology learning and tagging. Based on this analysis, it provides recommendations concerning the methodologies and implementations for the annotation of Web services developed in WP3 in SOA4ALL. In particular, this involves suggestions about the required input information, the learning approach itself, the level of automation and resulting ontology. These recommendations can be summarized as follows:

- The input information has semi-structured nature and should be complemented by patterns, which recognize terms based on the structure of the documents.
- The learning method should employ a pattern based approach, especially for the extraction of relations between concepts, in combination with machine learning. In addition, a domain expert should be involved, in order to improve the quality of the resulting ontology. Moreover, the developed method should enable the extraction of both concepts and relation instances.

8. References

1. (Agichtein and Gravano, 2000): E. Agichtein and L. Gravano. Snowball: Extracting Relations from Large Plain-Text Collections. In Proceedings of the 5th ACM International Conference on Digital Libraries (ACM DL), pages 85–94, 2000.
2. (Agirre et al., 2000): E. Agirre, O. Ansa, E. Hovy, and D. Martinez. Enriching Very Large Ontologies Using the WWW. In Workshop on Ontology Construction of the European Conference of A.I. (ECAI-00), 2000.
3. (Ahmad et al., 1994): K. Ahmad, A. Davies, H. Fulford, and M. Rogers. What is a term? The Semi-Automatic Extraction of Terms from Text. Amsterdam: John Benjamins Publishing Company, 1994.
4. (Alani et al., 2003a): Alani H., Sanghee K., Millard E.D., Weal J.M., Lewis P.H., Hall W., and Shadbolt N., Automatic Extraction of Knowledge from Web Documents, In: Proceeding of (HLT03), 2003.
5. (Alani et al., 2003b)**:** Alani H., Sanghee K., Millard E.D., Weal J.M., Lewis P.H., Hall W., and Shadbolt N., Web based Knowledge Extraction and Consolidation for Automatic Ontology Instantiation, In: Proceedings of the Workshop on Knowledge Markup and Semantic Annotation at the Second International Conference on Knowledge Capture (K-CAP 2003), Florida, USA, 2003.
6. (Alani et al., 2003c)**:** H. Alani, S. Kim, D.E. Millard, M.J. Weal, W. Hall, P.H. Lewis and N.R. Shadbolt (2003), "Automatic Ontology-Based Knowledge Extraction from Web Documents", IEEE Intelligent Systems, 18(1), pp. 14-21.
7. (Alfonseca et al., 2006): E. Alfonseca, M. Ruiz-Casado, M. Okumura and P. Castells. Towards Large-scale Non-taxonomic Relation Extraction: Estimating the Precision of Rote Extractors. In Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge – OLP 2006, pp. 49 – 56, Sydney, Australia, July 2006.
8. (Baroni and Bisi, 2004): M. Baroni and S. Bisi. Using cooccurrence statistics & the web to discover synonyms in a technical language. In Proceedings of the 4th International Conference on Language Resources and Evaluation, volume 5, pages 1725–1728, 2004.
9. (Begelman, G.; Keller, P. and Smadja, F., 2006): Automated Tag Clustering: Improving search and exploration in the tag space. WWW2006, May 22–26, 2006, Edinburgh, UK. Retrieved from: http://www.rawsugar.com/www2006/20.pdf
10. (Borgo et al., 1997): Borgo, S., Guarino, N., Masolo, C., and Vetere, G., Using a large linguistic ontology for internet based retrieval of object-oriented components, Proceedings of Conference on Software Engineering and Knowledge Engineering, IL, USA, 1997.
11. (Brooks, C.H. and Montanez, N., 2006): Improved Annotation of the Blogosphere via Autotagging and Hierarchical Clustering. WWW 2006, May 23–26, 2006, Edinburgh, UK. Retrieved from: http://www2006.org/programme/item.php?id=583
12. (Brewster et al., 2002): C. Brewster, F. Ciravegna and Y. Wilks (2002), "User-Centred Ontology Learning for Knowledge Management", 7th International Conference on Applications of Natural Language to Information Systems, Stockholm, June 27-28, 2002, Lecture Notes in Computer Science 2553, Springer Verlag.
13. (Buitelaar et al., 2004): P. Buitelaar, S. Handschuh, and B. Magnini, (eds.)

Proceedings of the ECAI04 Workshop on Ontologies, Learning and Population, 2004.

14. (Buitelaar et al, 2005): P. Buitelaar, P. Cimiano and B. Magnini. Ontology Learning from Text: Methods, Evaluation and Applications, IOS Press, 2005. ISBN: 1-58603-523-1
15. (Buitelaar et al., 2006):P. Buitelaar, P. Cimiano, B. Loos. Bringing the Gap between Text and Knowledge. Workshop on Ontology Learning and Population, 2006.
16. (Buitelaar et al., 2006b): P. Buitelaar, P. Cimiano, S. Racioppa and M. Siegel (2006), "Ontology-based Information Extraction with SOBA", In Proceedings of the International Conference on Language Resources and Evaluation, pp. 2321-2324. ELRA, May 2006.
17. (Corcho & Gómez-Pérez, 2000): Corcho O., and Gómez-Pérez, A., Evaluating Knowledge Representation and Reasoning Capabilities of Ontology Specification Languages, Proceedings of the ECAI 2000 Workshop on Applications of Ontologies and Problem-Solving Methods, Berlin, 2000.
18. (Craven et al., 2000): Craven M., DiPasquo D., Freitag D., McCallum A., Mitchell T., Nigam K., and Slattery S., Learning to Construct Knowledge Bases from the World Wide Web, Journal of Artificial Intelligence, vol. 118, no. 1/2, pp. 69-113, 2000.
19. (Cunningham et al., 2002a): H. Cunningham, D. Maynard, K. Bontcheva and V. Tablan. (2002), "Gate: an architecture for Development of Robust HLT Applications." Proceedings of ACL, 2002.
20. (Cunningham et al., 2002b): H. Cunningham, D. Maynard, K. Bontcheva and V. Tablan (2002), "GATE: a framework and graphical development environment for robust NLP tools and applications", In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics, Phil. USA.
21. (Cunningham et al., 2005): H. Cunningham, K. Bontcheva and Y. Li. (2005), "Knowledge management and human language: crossing the chasm", Journal of Knowledge Management vol.9, no.5, 2005, pp. 108-131.
22. (Damerau, 1993): F.J. Damerau. Evaluating domain-oriented multiword terms from texts. Information Processing and Management, 29(4):433–447, 1993.
23. (Downey et al., 2004): O. Downey, D. Etzioni, S. Soderland, and D. Weld. Learning Text Patterns for Web Information Extraction and Assessment. In Proceedings of the AAAI Workshop on Adaptive Text Extraction and Mining, 2004.
24. (Drozdzynski et al., 2004): W. Drozdzynski, H-U. Krieger, J. Piskorski, U. Schäfer and F. Xu. Shallow processing with unification and typed feature structures – foundations and applications. Künstliche Intelligenz, 1:17-23, 2004.
25. (Etzioni et al., 2004): Etzioni O., Kok S., Soderland S., Cagarella M., Popescu A.M., Weld D.S., Downey, Shaker T., and Yates A., Web-Scale Information Extraction in KnowItAll (Preliminary Results), in: Proceedings of the 13th International World Wide Web conference (WWW2004), New York, pp. 100-110, 2004.
26. (Etzioni et al., 2005): Etzioni O., Kok S., Soderland S., Cagarella M., Popescu A.M., Weld D.S., Downey, Shaker T., and Yates A., Unsupervised named-entity extraction from the Web: An experimental Study, Artificial Intelligence 165:91–

134, Elsevier, 2005.

27. (Faatz et al., 2002): A. Faatz and R. Steinmetz. Ontology Enrichment with texts from the WWW. In Semantic Web Mining 2nd Workshop at ECML/PKDD-2002, Helsinki, Finland, 2002.
28. (Fellbaum, 1998): C. Fellbaum (ed.) 1998. WordNet: An On-Line Lexical Database and Some of its Applications. Cambridge: MIT Press.
29. (Faure et al., 1998): Faure, D., Nedellec C., and Rouveirol, C., Acquisition of Semantic Knowledge using Machine Learning Methods: The System ASIUM, Technical Report number ICS-TR-88-16, Laboratoire de Recherche en Informatique, Inference and Learning Group, Universite Paris Sud, 1998.
30. (Faure and Poibeau, 2000): Faure D., and Poibeau, T., First experiments of using semantic knowledge learned by ASIUM for information extraction task using INTEX, Proceedings of the ECAI 2000 Workshop on Ontology Learning (OL'2000), 2000.
31. (Fortuna et al., 2005): B. Fortuna, D. Mladevic, and M. Grobelnik. Visualization of Text Document Corpus. In ACAI 2005 Summer School, 2005.
32. (Frantzi et al., 2000): K. Frantzi, S. Ananiadou, and H. Mima. Automatic recognition of multi-word terms: The c-value/nc-value method. International Journal on Digital Libraries, 3(2):115–130, 2000.
33. (Gruber, 1994): T. Gruber. Towards principles for the design of ontologies used for knowledge sharing. Int. J. of Human and Computer Studies, 43:907–928, 1994.
34. (Gruber, T. 2007). Ontology of folksonomy: A mashup of apples and oranges. Intl Journal on Semantic Web & Information Systems, 3(2), 2007. Available at: http://tomgruber.org/writing/ontology-offolksonomy.htm.
35. (Golder, S., & Huberman, B., 2006): Usage patterns of collaborative tagging systems. Journal of Information Science, 32(2), 198-208. Retrieved November 29, 2006, from http://www.hpl.hp.com/research/idl/papers/tags/tags.pdf.
36. (Gómez-Pérez and Manzano-Macho, 2003): A. Gómez-Pérez, D. Manzano-Macho. A survey of ontology learning methods and techniques. Onto-web IST Project, Deliverable 1.5: http://ontoweb.aifb.uni-karlsruhe.de/Members/ruben/Deliverable%201.5
37. (Gupta et al., 2002): K.M. Gupta, D. Aha, E. Marsh, and T. Maney. An Architecture for engineering sublanguage WordNets. In Proceedings of the First International Conference On Global WordNet, Mysore, India: Central Institute of Indian Languages, pages 207–215, 2002.
38. (Guy, M., & Tonkin, E., 2006): Folksonomies: Tidying up Tags? D-Lib Magazine, 12(1). Retrieved November 10, 2006, from http://www.dlib.org/dlib/january06/guy/01guy.html.
39. (Haase and Stojanovic, 2005): P. Haase and L. Stojanovic. Consistent Evolution of OWL Ontologies, pages 182–197. Springer, LNCS 3532, 2005.
40. (Haase et al, 2005): P. Haase, F. Van Harmelen, Z. Huang, H. Stuckenschmidt, and Y. Sure. A Framework for Handling Inconsistency in Changing Ontologies. In Proceedings of the 2005 International Semantic Web Conference, ISWC2005, 2005.
41. (Haase and Volker, 2005): P. Haase and J. Volker. Ontology Learning and

Reasoning - Dealing with Uncertainty and Inconsistency. In Proceedings of the Workshop on Uncertainty Reasoning for the Semantic Web (URSW), 2005.

42. (Hahn and Marko, 2002): Hahn U., and Marko, K. G., Ontology and Lexicon Evolution by Text Understanding, Proceedings of the ECAI 2002 Workshop on Machine Learning and Natural Language Processing for Ontology Engineering (OLT'2002), Lyon, France, 2002.
43. (Hearst, 1992): M.A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. In Proceedings of the 14th International Conference on Computational Linguistics, Nantes, France, 1992.
44. (Harris, 1968): Z. Harris. Mathematical Structures of Language. John Wiley & Sons, 1968.
45. (Hindle, 1990): D. Hindle. Noun classification from predicate-argument structures. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, pages 268–275,1990.
46. (Iria et al., 2006): J. Iria, C. Brewster, F. Ciravegna and Y. Wilks (2006), "An Incremental Tri-Partite Approach To Ontology Learning", In the 5th International Conference on Language Resources and Evaluation, Genoa, 24-25-26 May 2006.
47. (Iwanska et al., 2000): L.M. Iwanska, N. Mata, and K. Kruger. Fully Automatic Acquisition of Taxonomic Knowledge from Large Corpora of Texts, pages 335–345. MIT/AAAI Press, 2000.
48. (Kietz et al., 2000): J.U. Kietz, A. Maedche, and R. Volz. A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet. In Proceedings of the ECAW-2000 Workshop "Ontologies and Text", Juan-Les-Pins, France, 2000.
49. (Kim H. L, et al., 2007): Simple algorithms for representing tag frequencies in the scot exporter. In IAT, pages 536–539. IEEE Computer Society, 2007.
50. (Krauthammer and Nenadic, 2004): M Krauthammer and G. Nenadic. Term identification in the biomedical literature. Journal of Biomedical Informatics, 37:512–526, 2004.
51. (Landauer and Dumais, 1997): T.K. Landauer and S.T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychological Review, 104:211–240, 1997.
52. (Lesk, 1986): M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In ACM SIGDOC '86, The Fifth International Conference on Systems Documentation, 1986.
53. (Lin and Pantel, 2001): D. Lin and P. Pantel. Induction of semantic classes from natural language text. In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 317–322, 2001.
54. (Lin and Pantel, 2001b): D. Lin and P. Pantel. Dirt - Discovery of Inference Rules from Text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 323–328, 2001.
55. (Lin and Pantel, 2002): D. Lin and P. Pantel. Concept discovery from text. In Proceedings of the International Conference on Computational Linguistics (COLING), pages 577–583, 2002.
56. (Lisi, 2005): F.A. Lisi. Principles of Inductive Reasoning on the Semantic Web: A Framework for Learning in AL-Log. In F. Fages and S. Soliman (Eds.), Principles and Practice of Semantic Web Reasoning, Series: Lecture Notes in Computer

Science, Vol. 3703, 118-132, Springer: Berlin, 2005.

57. (Maedche and Staab, 2000a): A. Maedche and S. Staab. Semi-Automatic Engineering of Ontologies from Text. In Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering, 2000.
58. (Maedche and Staab, 2000b):A. Maedche and S. Staab. Discovering Conceptual Relations from Text. In Proceedings of ECAI 2000, IOS Press, Amsterdam, 2000.
59. (Maedche et al, 2001): A. Maedche, S. Staab, C. N´edellec, and Ed Hove, editors. IJCAI' 01 Workshop on Ontology Learning, volume http://CEUR-WS.org/Vol-38/ CEUR, 2001.
60. (Maedche and Staab, 2001): A. Maedche, and S. Staab, Ontology learning for the Semantic Web, IEEE journal on Intelligent Systems, Vol. 16, No. 2, 72-79, 2001.
61. (Maedche and Staab, 2002): A. Meadche and S. Staab. Measuring Similarity Between Ontologies. In Proceedings of the European Conference on Knowledge Acquisition and Management (EKAW), pages 251–263, 2002.
62. (Marlow, et al., 2006): HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, ToRead. Hypertext 06. Retrieved from http://www.danah.org/papers/Hypertext2006.pdf.
63. (Mejias, U.A., 2004): Bookmark, Classify and Share: A mini-ethnography of social practices in a distributed classification community. Blog posting. Retrieved January 5, 2007, from http://ideant.typepad.com/ideant/2004/12/a_delicious_stu.html.
64. (Mika, P., 2005): Ontologies are us: A unified model of social networks and semantics. Proceedings of the 4th International Semantic Web Conference (ISWC 2005), LNCS 3729, Springer-Verlag, 2005. Retrieved from: http://www.cs.vu.nl/~pmika/research/papers/ISWC-folksonomy.pdf
65. (Morin, 1999): E. Morin. Automatic Acquisition of Semantic Relations Between Terms from Technical Corpora. In Proceedings of the Fifth International Congress on Terminology and Knowledge Engineering - TKE'99, 1999.
66. (Navigli and Velardi, 2005): R. Navigli and P. Velardi. Structural Semantic Interconnections: a knowledge-based approach to word sense disambiguation. Special Issue-Syntactic and Structural Pattern Recognition, IEEE TPAMI, Volume: 27, Issue: 7, 2005.
67. (Navigli and Velardi, 2006a): R. Navigli and P. Velardi. Enriching a Formal Ontology with a Thesaurus: an Application in the Cultural Heritage Domain. In Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge – OLP 2006, pp. 1 – 9, Sydney, Australia, July 2006.
68. (Navigli and Velardi, 2006b): R. Navigli and P. Velardi. Ontology Enrichment Through Automatic Semantic Annotation of On-Line Glossaries. In Proceedings of the EKAW 2006 - 15th International Conference on Knowledge Engineering and Knowledge Management - Managing Knowledge in a World of Networks, pp. 126-140, Podebrady, Czech Republic, October 2-6, 2006.
69. (Newman R., Danny Ayers, and Seth Russell, 2005): Tag ontology, December 2005. Available at: http://www.holygoat.co.uk/owl/redwood/0.1/tags/.
70. (Quinlan, 1990): J.R. Quinlan. Learning logical definitions from relations. In Machine Learning 5 (1990) 239–266.

71. (Roux et al., 2000): Roux, C., Proux, D., Rechenmann F., and Julliard, L., An Ontology Enrichment Method for a Pragmatic Information Extraction System gathering Data on Genetic Interactions, Proceedings of the ECAI 2000 Workshop on Ontology Learning (OL'2000), 2000.

72. (Salton et al., 1975): A. Saltion, G.Wong and C.S. Yang. A vector space model for automatic indexing. Communications of the ACM, 18(11):613–620, 1975.

73. (Salton, G.; McGill, M.J., 1983): Modern information retrieval. New York, McGraw Hill.

74. (Schütze, 1993): Hinrich Schütze. Word space. In Advances in Neural Information Processing Systems 5, 1993.

75. (Schutz and Buitelaar, 2005): A. Schutz and P. Buitelaar. RelExt: A Tool for Relation Extraction from Text in Ontology Extension. In Proceedings of the 4th International Semantic Web Conference, 2005.

76. (Shamsfard and Barforoush, 2002): Shamsfard M., and Barforoush, A. A., (a) An Introduction to HASTI: An Ontology Learning System, Proceedings of 6th Conference on Artificial Intelligence and Soft Computing (ASC'2002), Banff, Canada, June, 2002.

77. (Shamsfard, 2003): M. Shamsfard, Designing the ontology learning Model, Prototyping in a Persian Text Understanding System, Ph.D. Dissertation, Computer Engineering Dept., AmirKabir University of Technology, Tehran, Iran, Jan. 2003.

78. (Shamsfard and Barforoush, 2003): M. Shamsfard, and A. Abdollahzadeh Barforoush. The state of the art in ontology learning: a framework for comparison. Knowl. Eng. Rev. 18, 4 (Dec. 2003), 293-316. DOI= http://dx.doi.org/10.1017/S0269888903000687

79. (Shamsfard and Barforoush, 2004): Shamsfard M., and Barforoush, A. A. Learning Ontologies from Natural Language Texts. In International Journal .of Human-Computer Studies, No. 60, pp. 17-63, Jan.2004

80. (Shaw, B., 2005): Semi definite Embedding Applied to Visualizing Folksonomies. Project Proposal. December, 2005. Retrieved from: http://www.metablake.com/advml/adv-ml-project.pdf

81. (Schulte im Walde, 2000): S. Schulte im Walde. Clustering Verbs Semantically According to their Alternation Behaviour. In Proceedings of the 18th International Conference on Computational Linguistics (COLINGS), pages 747–753, 2000.

82. (Sowa, 2000): Sowa, J. F., Knowledge Representation: Logical, Philosophical and Computational Foundations, Brooks/Cole, 2000.

83. (Soderland et al., 1995): Soderland, S., Fisher, D., Aseltine, J., Lehnert, W., Issues in Inductive Learning of Domain-Specific Text Extraction Rules, Proceedings of the IJCAI 95 Workshop on Approaches to Learning for Natural Language Processing, 1995.

84. (Specia and Motta, 2006): L. Specia and E. Motta. A hybrid approach for extracting semantic relations from texts. In Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge – OLP 2006, pp. 57 – 64, Sydney, Australia, July 2006.

85. (Suchanek et al. 2006): F.M. Suchanek, G. Ifrim and G, Weikum. LEILA: Learning to Extract Information by Linguistic Analysis. In Proceedings of the 2nd Workshop

on Ontology Learning and Population: Bridging the Gap between Text and Knowledge – OLP 2006, pp. 18 – 25, Sydney, Australia, July 2006.

86. (Suryanto & Compton, 2000): Suryanto H., Compton, P., Learning classification taxonomies from a classification knowledge based system, Proceedings of the ECAI 2000 Workshop on Ontology Learning (OL'2000), 2000.
87. (Thompson&Mooney, 1999): Thompson, C. A., Mooney, R. J., Automatic Construction of Semantic Lexicons for Learning Natural Language Interfaces, Proceedings of 16th National Conference on Artificial Intelligence (AAAI'99), 487-493, Orlando, Florida, 1999.
88. (Turney, 2001): P. D. Turney. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In Proceedings of the Twelfth European Conference on Machine Learning, pages 491–502, Freiburg, Germany, 2001.
89. (Volker et al., 2005): J. Volker, D. Vrandecic, and Y. Sure. Automatic Evaluation of Ontologies (AEON). In Proceedings of the 4th International Semantic Web Conference, 2005.
90. (Xu, Z. et al., 2006): Towards the Semantic Web: Collaborative Tag Suggestions. WWW2006, May 22–26, 2006, Edinburgh, UK. Retrieved from: http://www.rawsugar.com/www2006/13.pdf
91. (Yarowsky, 1992): D. Yarowsky. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In COLING-92, Nantes, 1992.
92. (Zavitsanos et al., 2006): E. Zavitsanos, G. Paliouras, and G. Vouros. Ontology Learning and Evaluation: A survey. Technical report, DEMO-2006-3), NCSR Demokritos, Athens, Greece, 2006.