Enhancing the Ontological Knowledge through Ontology Population and Enrichment

Alexandros G. Valarakos†‡, Georgios Paliouras†, Vangelis Karkaletsis†, and George A. Vouros‡

† Software and Knowledge Engineering Laboratory
Institute of Informatics and Telecommunications,
National Centre for Scientific Research "Demokritos"
153 10 Ag. Paraskevi, Athens, Greece
Tel: +30 210 6503197, Fax: +30 210 6532175
{alexv, paliourg, vangelis}@iit.demokritos.gr

‡ Department of Information and Telecommunication Systems
Engineering, School of Sciences, University of the Aegean
83200, Karlovassi, Samos, Greece
Tel: +30 2 273 82226, Fax: +30 2 273 82009
georgev@aegean.gr

Abstract. Ontologies are widely used for capturing and organizing knowledge of a particular domain of interest. This knowledge is usually evolvable and therefore an ontology maintenance process is required to keep the ontological knowledge up-to-date. We proposed an incremental ontology maintenance methodology which exploits ontology population and enrichment methods to enhance the knowledge captured by the instances of the ontology and their various lexicalizations. Furthermore, we employ ontology learning techniques to alleviate as much as possible the intervention of human into the proposed methodology. We conducted experiments using the CROSSMARC ontology as a case study evaluating the methodology and its partial methods. The methodology performed well enhancing the ontological knowledge to 96.5% from only 50% .

1. Introduction

The World Wide Web is the richest repository of information, whose semantics are oriented to humans rather than to machines. The enrichment of the Web with semantic annotations (metadata) is fundamental for the accomplishment of the Semantic Web [4], and is currently performed manually [11] or semi-automatically [7] [8] [10]. Semantic annotations associate information with specific entities within the domain of interest, aiming to facilitate a semantic-based interpretation of content by restricting their formal models of interpretation through ontologies. Domain entities are represented as instances of concepts in ontologies. A domain ontology captures knowledge in a static way, as it is a snapshot of knowledge concerning a

specific domain from a particular point of view (conceptualization), in a specific time-period.

On the other hand, ontologies have the potential to organize and centralize knowledge in a formal, machine and human understandable way, making themselves an essential component to many knowledge-intensive services. However, due to changes concerning knowledge-related requirements and depending on the evolutionary tendencies of the domain of interest, a domain ontology might contain incomplete or out-of-date knowledge regarding its instances. For example, an ontology that has been constructed for the domain "laptop descriptions" last year will miss the latest processor types used in laptops. Moreover, the different surface appearance (lexicalization) of an instance, which can appear through the time, restricts the knowledge a domain ontology intends to capture. For example, the ignorance that the instance "Intel Pentium 3" can be appeared as "P III" is a serious knowledge leak for the modeling of the "laptop descriptions" domain. Thus, the maintenance of its captured knowledge is crucial for the performance of the application that uses it. Maintaining ontological knowledge through population and enrichment is a time-consuming, error prone and labor-intensive task when performed manually. Ontology learning can facilitate the ontology population and enrichment process by using machine learning methods to obtain knowledge from data.

In the context of ontology maintenance we employ ontology population and enrichment methods to enhance the knowledge captured in a domain ontology. We focus on the maintenance of the ontological instances and their various lexicalizations. We identify new instances for concepts of a domain ontology and add them into it (ontology population method). Moreover, we acquire a non-taxonomic relationship between instances (ontology enrichment method) that captures their different lexicalizations avoiding the existence of duplicate ontological instances. The latter is a problem that has not been given sufficient attention [19]. We integrate those methods in a proposed incremental methodology, which comprises the ontology population and enrichment methods aiming at the enhancement of the knowledge contained into the ontology concerning its ontological instances and their different lexicalizations. Furthermore, we employ machine learning methods for alleviating as much as possible the role of human into the ontology maintenance process.

In the next section we give information concerning the ontology of the information integration system CROSSMARC[1], which we used for our experiments for describing a case study. Then we present our incremental ontology population and enrichment methodology in section 3. Section 4 describes the experimental results on the population and enrichment of the domain ontology concerning laptop descriptions. Finally, we present the related work in section 5 and we conclude in section 6.

[1] CROSSMARC is an R&D project under the IST Programme of the European Union (IST-2000-25366). http://www.iit.demokritos.gr/skel/crossmarc

2 The CROSSMARC Ontology

The ontology that was used in our case study describes laptop products and has been manually constructed using the Protégé-based[2] management system developed in the context of the CROSSMARC project and will be public available at the CROSSMARC web site. The ontology was implemented in an xml dialect and consists of '*part-of*' relationship, which link the main concept, namely *laptop*, to its parts (e.g. *processor*, *manufacturer*, *screen*, *price* etc.) Additionally, there is a '*has attribute'* property for each concept (e.g. concept "*processor*" has attribute "*processor name"*) which its range could be a string or a numeric data-type followed by its measurement unit (e.g. the concept "Hard Disk" has attribute "capacity" which its value space is defined by an integer datatype followed by a measurement unit-the literal "G.B."). Also, there are constraints on the range of numerical data-types which are defined by a minimum and maximum value. Moreover, there is an '*instance-of*' relationship that denotes the instances of the concepts, e.g. '*IBM'* and *'Toshiba* instantiate the concept 'Manufacturer Name'. Furthermore, a lexical '*synonymy*[3]*'* relationship associates the appropriate different surface appearances of an instance.

3 Knowledge Enhancement: Methodology

As we have already pointed out, in the context of our ontology maintenance methodology, we focus on the increase of ontological knowledge concerning the instances that exists in a domain of interest and of their lexical synonyms (different lexicalizations of an instance). Thus, we have to accomplish two subtasks: firstly to populate the ontology with new instances, and secondly, to acquire lexical synonymy relationships between the different lexicalizations of an instance. We employ an ontology population and enrichment method to deal with the first and second subtask, respectively.

The key idea is that we can keep the ontology instances and their lexical appearances up-to-date in a semi-automatic way, by periodically re-training an information extraction system using a corpus that contains the target knowledge and have been partially annotated (scattered annotations) using the already known instances captured in the ontology. The corpus is constructed gradually to secure the existence of new instances and their lexical synonyms, depending on the rate of domain instances' updates i.e. the ratio of the new instances to the already known that exist in the corpus for a specific time interval. The instances already in the ontology, as well as their lexical synonyms are used for annotating corpus' documents employing domain specific disambiguation techniques in order to provide semantically consistent annotations (semantic consistency problem) i.e. according to our ontology the string "Intel Pentium" is annotated only if it refers to processor name

[2] Protégé Web Site: http://protege.stanford.edu/

[3] The meaning of this word is overridden; it refers mainly to the surface appearance of an instance rather to its meaning.

and not to something other (e.g. company name). These scattered annotations constitute the training dataset that will be used by the information extraction system.

3.1 Incremental Ontology Population and Enrichment

The incremental ontology population and enrichment methodology proposed, iterates through four stages:

1. ***Ontology-based Semantic Annotation***. The instances of the domain ontology are used to semantically annotate a domain-specific corpus in an automatic way. In this stage disambiguation techniques are used exploiting knowledge captured in the domain ontology.
2. ***Knowledge Discovery***. An information extraction module is employed in this stage to locate new ontological instances. The module is trained, using machine learning methods, on the annotated corpus of the previous stage.
3. ***Knowledge Refinement***. A compression-based clustering algorithm is employed in this stage for identifying lexicographic variants of each instance supporting the ontology enrichment.
4. ***Validation and Insertion***. A domain expert validates the candidate instances that have been added in the ontology.

Figure 1 depicts the above methodology which is presented in more detail in the following subsections.

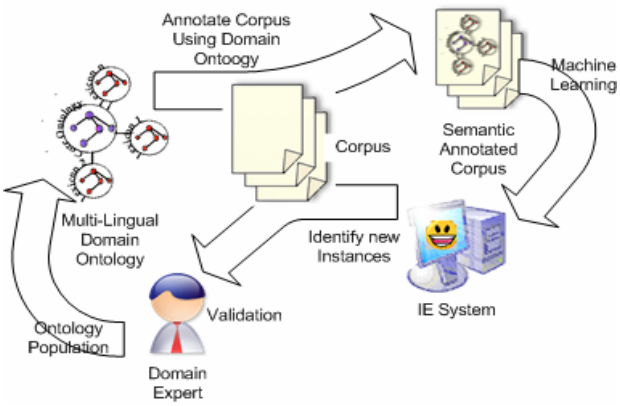

Fig. 1: Overall Method for Ontology Maintenance

3.1.1 Ontology-based Semantic Annotation

The aim of this stage is to annotate a corpus with existing concept instances. This instance-based method differs from the semi-automated semantic annotation that has

been proposed in the literature, as it intends to automatically annotate a document with metadata derived explicitly from the ontology at hand. Other methods (appear in section 5) can be characterized as concept-based as they intend to annotate all the potential instances that can be found in a corpus and belong to a particular concept. These methods usually exploit context-typed information using information extraction methods. Obviously, the instance-based semantic annotation is faster as it does not need to identify new instances but requires disambiguation techniques as the latter does as well. On its own, this method is sufficient when our knowledge about a domain is closed or when we are interested only in the known concept instances.

The semantic annotation of the corpus is currently performed by a string matching technique that is biased to select the maximum spanning annotated lexical expression for each instance. One problem with this method is the identification of *properties*, whose range of values is a numerical datatype followed by the corresponding measurement unit, e.g. dates, age, capacity. For example, the numeric string *"32"* could be an instance of ram memory or hard disk capacity. Those ambiguities are resolved by the exploitation of the measurement units (knowledge encoded in ontology) e.g. if the *"32"* is being followed by the string *"kb"* then it is a ram memory's instance and if is being followed by the string *"GB"* then it is an instance of the hard disk capacity. Beyond exploiting measurement units, properties are also identified by special rules that enhance string matching techniques by using again knowledge encoded in the ontology, such as the valid range of values that a property can take. For example, RAM capacity values range in a set that is different from the one that the Hard disk capacity ranges. We encode such knowledge in the definition of the concept and use it to resolve the ambiguities.

3.1.2 Knowledge Discovery

At this stage, in the context of ontology population, we aim to identify new ontological instances that are not included in the ontology. For this purpose, we use Hidden Markov Models (HMMs) to train an information extraction module for locating new ontological instances. We train a single HMM for each set of ontological instances that belong to a particular concept, as proposed in [1] and [2]. HMMs exploit only tokens, intending to capture the context in which the instances of a particular concept appear in. The structure of each HMM is set by hand. The model parameters are estimated in a single pass over the training dataset by calculating ratios of counts (maximum likelihood estimation). At runtime, each HMM is applied to one document in the corpus, using the Viterbi procedure to identify matches.

The first stage of instance-based annotation provides training examples to the HMM. This ontology-driven machine learning approach differs from the classical supervised methods as it does not use human-provided training examples but examples provided by the domain ontology. After the training phase, the trained information extraction module is capable of recognizing new ontological instances for the concepts on which it has been trained. The extracted ontological instances constitute the set of candidate instances that will be validated by the domain expert.

3.1.3 Knowledge Refinement

This stage aims to reduce the amount of work required by the domain expert, by identifying different lexicalizations of the same ontological instance. For example, the processor name *'Pentium 2'* can be written differently as *'Pentium II'*, *'p2'*, *'P II'* or *'Intel Pentium 2'*.

The identification of different lexicalizations of existing instances is performed by a novel compression-based clustering algorithm, named COCLU [13].This algorithm is based on the assumption that different lexicalizations of an instance use a common set of 'core' characters. Therefore, lexicalizations that are 'close' to this set are potential alternative appearances of the same instance, while those that are 'far' from this set are potentially related to a different instance.

COCLU is a partition-based clustering algorithm which divides the data into several subsets and searches the space of possible subsets using a greedy heuristic. Each cluster is represented by a model, rather than by the collection of data assigned to it. The cluster's model is realized by a corresponding Huffman tree which is incrementally constructed, as the algorithm dynamically generates and updates the clusters by processing one string (instance's lexicalization) at a time. The algorithm employs a new score function that measures the compactness and homogeneity of a cluster. This score function is defined as the difference of the summed length of the coded string tokens that are members of the cluster, and the length of the same cluster updated with the candidate string. The score function groups together strings that contain the same set of frequent letters according to the model of a cluster.

The use of COCLU is two-fold. Firstly, it can be used as a classifier which assigns a candidate string to the appropriate cluster-concept. A cluster is defined by an ontological instance and its synonyms. This scenario takes place when we want to discover synonyms only for the existing ontological instances. Additionally, COCLU can be used for discovering new clusters-concepts beyond those denoted by the existing ontological instances. In this way, COCLU can discover new instances in an unsupervised way.

3.1.4 Validation and Insertion

At this stage a domain expert should validate the candidate ontological instances as well as their synonyms that have been added to the ontology. At the end of this phase the method starts again from the first stage until no more changes are possible.

4. Experimental Results

We evaluated the performance of the incremental ontology population and enrichment methododology presented in section 3 on the laptop domain of the CROSSMARC project. Our intention is to evaluate the proposed method in acquiring all the knowledge that exists in the given corpus. We conducted experiments concerning the instance and concept-based semantic annotation as well as their combination to prove

that is possible the innovative approach of training an information system exploiting the instances of a domain ontology, the knowledge captured in the ontology and a given domain-specific corpus, using automatically produced training examples. Moreover, the combination of the two methods (concept-based & instance-based) is possible to provide as with the appropriate knowledge. We also evaluated the COCLU algorithm in discovering instances that participate in the lexical synonymy relationship. Furthermore, we measured the performance of the method in discovering new ontological instances in its incremental mode.

The CROSSMARC laptop ontology covers all four languages examined in the project. However, we concentrated in the English instantiation, which consists of 119 instances. The corpus for English consists of 100 Web pages containing laptops' descriptions and is public available. The corpus processing was done using the text engineering platform Ellogon [3]. The proposed method requires the pre-processing of the corpus only from a tokenizer which identifies text tokens (e.g., words, symbols, etc.) in the Web pages and characterizes them according to a token-type tag set which encodes graphological information (e.g. the token is an English upper case word).

4.1 Knowledge Discovery

In order to investigate the tolerance of the method to the number of examples available for training the HMM, we performed three separate annotations of the training corpus using 75%, 50% and 25% of the initial ontology, respectively. These subsets represent the portion of the ontological knowledge that already exists in the corpus, as the corpus is constructed gradually. Therefore, the remaining documents in the corpus contain concept instances and their lexical synonyms that should be acquired. These subsets were constructed based on the evolution of the instances in time, thus simulating the real use of the methodology. For instance, in the laptop domain, "Pentium" is a predecessor of "Pentium 3". Thus "Pentium" was selected to participate in the 25% of the initial ontology.

Applying the proposed methodology, we:

- annotated the corpus using the ontology
- used the annotated corpus to train the HMMs and
- applied the trained HMMs on the corpus to discover new ontological instances.

In case that annotations comprise inexact matches, the more precise ontology-based annotation was preferred over the HMM annotation. For example, when the same offset has been annotated by the HMM-based and ontology-based method the one that dominates and remains is the ontology-based method. The same happens when we have overlapping annotation offset between these methods.

The performance of the HMM-based and ontology-based annotation method as well as their combination was evaluated separately using the precision and recall measures. Table 1 shows average results for 5 concepts chosen from CROSSMARC ontology and for an ontology size of 75%, 50% and 25%, respectively. We have chosen only the concepts for which new instances came out in a short period of time. The first row shows the results of applying the ontology-based method using the initial ontology. The second row shows the results of the trained HMM and the last row shows the results of their combination. It is worth noting that these results are

limited to the first iteration of the methodology and at least one occurrence of an instance or a lexical synonym in the corpus is enough to indicate its successful discovery-annotation. However, the different sizes of the ontology used simulate, in a way, the iteration.

% ontology	*75*		*50*		*25*	
Method	**P (%)**	**R (%)**	**P (%)**	**R (%)**	**P (%)**	**R (%)**
Ontology-based	100,0	66,1	100,0	50,0	100,0	24,5
HMM-based	69,2	65,5	62,3	47,7	68,0	26,3
Combination	74,0	76,0	67,0	57,4	71,3	33,1

Table 1: Semantic annotation results in the first iteration.

Examining the annotation using the original ontology, precision (**P**) was perfect, as expected. On the other hand, recall (**R**) was directly affected by the coverage of the ontology. The precision of the HMM-based annotation varied between 62.3% and 69.2%, while its recall was comparable to that of the ontology. However, the combination of the two methods performed better in terms of recall, as the HMMs provided new instances not included in the ontology. Furthermore, the precision of the combined approach was higher than that of the HMM-based annotation. Thus, the combination of the two methods seems like a viable approach to generating potential new instances for the ontology.

4.2 Knowledge Refinement

Table 2 measures the ability of COCLU to find lexical variations of correct concept instances. The grouping of instances can be considered to be classes whose members are different lexicalizations of an instance, hence are linked with the synonymy relationship. The experimental approach is similar to that presented above, i.e., we hide a number of randomly selected synonyms that are being linked with the instances of the selected concepts of the ontology and ask COCLU to classify them. The accuracy of the algorithm decreases as the number of hidden synonyms increases. However, it is encouraging that cluster's size can be further reduced to almost half of it without any loss in accuracy.

Instance Reduction (%)	Correct	Accuracy (%)
90	3	100
80	11	100
70	15	100
60	19	100
50	23	95,6
40	29	96,5
30	34	94,1

Table 2: Instance matching results for COCLU

We have also measured COCLU's ability to discover new ontological instances beyond the existing ones. These new ontological instances constitute new classes

(herein they are named clusters) that their members have significantly different lexicalizations of the concept instances. To evaluate this method we hid incrementally from one to all clusters that have been constructed manually in the target ontology, measuring the algorithm's ability to discover the hidden clusters. We explored all possible combinations of hidden clusters. In all trials, COCLU has re-generated all the hidden clusters. However, it was consistently splitting two of the clusters into smaller sub-clusters. In standard information retrieval terms, the recall of the algorithm was 100%, as it managed to generate all the required clusters, while its precision was 75%.

4.3 Method in Incremental Mode

In this experiment, we evaluated the improvement of the results as the methodology is iteratively applied (incremental mode) to the same corpus. Again we used 25%, 50% and 75% of the instances that exist in the target ontology as a starting point. Table 3 provides the results obtained by the experiment. Each row denotes the percentage of the initial concept instances used. Columns provide the number of the initial and target instances, as well as the number of instances annotated by the initial ontology (0^{th} iteration) and by the system in each subsequent iteration. We do not count the multiple occurrences of an instance in the corpus as the number of extracted instances but only one occurrence is enough to characterize as successful the discovery of an instance. Also, it is worth noting that the method locates all the instances that are able to be discovered until the 2^{nd} iteration. After this iteration no further improvement is noticed.

Initial ontology	Initial instances	Target instances	0^{th} Iteration	1^{st} Iteration	2^{nd} Iteration	Final Coverage
25%	15	58	23	7	3	82.7%
50%	28	58	20	5	3	96.5%
75%	40	58	14	3	0	98.7%

Table 3: Evaluation of the overall method

The number of iterations required to retrieve most of the instances depends on the size of the initial concept instances, but is generally small. Starting with only 50% of the target instances the method succeeds to populate the ontology increasing its coverage from 48.3% to the 96.5% of the target instances in 2 iterations.

It is worth noticing that a study on the evolution rate of a domain can indicate us with the exact time period in which the proposed methodology should be applied for keeping up-to-date the ontological knowledge. If the ontology contains at least the 50% of the instances that exist in the corpus, the methodology secures 96.5% coverage of the knowledge contained in the corpus.

5 Related Work

Ontology population can be characterized as the evolution of semantic lexicon learning [Riloff] task as the main difference is the formalism of the resource (dictionary of words with semantic category labels or an ontology) that accumulates the instances and will be populated with new one. The richer representation power of an ontology can bootstrap the task of learning exploiting knowledge encoded in it as this happens at the semantic annotation stage of our methodology.

The task of semantically annotating a corpus from several resources (ontologies, thesaurus, semantic lexicons or combination of them) has been researched by many works. As stated in section 3.1.1 we divide this task into concept and instance based approaches. The latter one concerns the recognition of all the instances that exist in the ontology and appear in the corpus [Paul]. A more sophisticated extension of this method usually uses disambiguation techniques to support the correct sense attribution of an ontological instance according to the ontology used [Dill]. The concept based approach, aiming to discover new instances beyond the one exist in the ontology, employs information extraction techniques [Dingli:Automatic Semantic] [ref]. This approach is one step before be characterized as ontology population approach. Its missing part is the insertion of new concept instances under the appropriate concept of the ontology.

Various approaches based on information extraction methods have already been used for ontology population. Most of them uses information extraction systems to locate (mark up) the concept instances relying on manually annotated corpus [MnM:7]. Some of them face the semantic consistency problem allowing the human to evaluate the training examples that will feed the information extraction system [ciravegnia:User Centered]. In contrast to Ciravegnia's work, our work relies entirely on the automatic creation of the training corpus exploiting the ontology-based annotation method which uses the knowledge encoded in the ontology when semantic ambiguities rise. Furthermore, we deal with the identification of typographic variations of an instance and their population. Those typographic variations are used in the semantic annotation stage as constitute part of our ontological knowledge. In [10:Popov] the problem is posed as a named entity recognition problem that uses linguistic analysis processes and manual crafted rules for identifying instances in documents. Although, this work intend to identify domain-independent name entities, the use of manual crafted rules are biased by the format of the documents. Moreover, all these approaches pre-process the corpus with various linguistic processes (e.g. part-of-speech tagging, morphological analysis, chunking etc.) whereas our approach uses only a tokenizer, hence it is faster.

The problem of the existence of instances that refer to the same entity is dealt in [9] using heuristic comparison rules. The same problem is being met in the database community as the existence of duplicate records [17] and in the Natural Language Processing community as the name matching problem [18]. We deal with this problem developing a machine learning algorithm named COCLU which exploits character typed information supporting a particular lexical synonymy relationship that is being implicitly used by many applications.

6 Conclusions

We have presented an incremental methodology for ontology maintenance, exploiting ontology population and enrichment techniques. Following the objective of CROSSMARC project for quick adaptation to new domains and languages we devised a methodology for efficient maintenance of the CROSSMARC ontology. This is crucial in the context of CROSSMARC, as the ontology has a key role in the functions of most of CROSSMARC components.

The proposed methodology uses the ontology for automatically annotating a domain specific corpus. The annotated corpus is then used to train an information extraction system. The trained system identifies new candidate instances which are processed by a compression-based algorithm in order to discover lexical synonyms among them. Finally the candidate instances and the proposed lexical synonyms are validated by a domain expert. The method iterates until no new instances are being found.

We conducted experiments using the ontology and the corpus of a CROSSMARC domain (laptops' descriptions) in one of the project languages (English). We evaluated each stage of the proposed methodology separately as well as the overall methodology. The initial results obtained are encouraging. The coverage of the ontology increased to 96.5% starting from a coverage of only 50%. Also, the clustering algorithm COCLU performed quite well assigning with 95.6% success a candidate instance to the correct cluster indicating the new instance that participates in the lexical synonymy relationship. It also managed to discover new pair of instances that are associated with the lexical synonymy relationship.

Concluding, the combination of the ontology-based method with the HMM-based annotation method gave very good results on a corpus of web pages with semi-structured content exploiting only token type information. Also, the incremental mode of the method indicates that the 50% of the instances that exists in the corpus is adequate enough for acquiring the 96.5% of the total instances.

In addition to the need for further experimentation of the proposed method, we plan to do large-scale experiments. The disambiguation technique used in our approach, which is driven by the measurement units, proved to work well but further research on this direction should be done for establishing this approach reliable enough. Also, we plan to investigate the discovery of instances realizing other types of synonymy relationships by extending the COCLU algorithm to identify them. Furthermore, we plan to support the semi-automatic maintenance of ontologies implemented in OWL, providing a platform that will centralize all these supportive tools.

References

[1] Freitag, D., McCallum, A.K., Information Extraction using HMMs and shrinkage, In Proceedings of AAAI-99 Workshop on Machine Learning for Information Extraction, pp. 31-36 (1999).

[2] Seymore, K., McCallum A.K., Rosenfeld, R., Learning hidden Markov model structure for Information Extraction, *Journal of Intelligent Information Systems* 8(1): 5-28, (1999).

[3] Petasis G., V. Karkaletsis and C.D. Spyropoulos. "Cross-lingual Information Extraction from Web pages: the use of a general-purpose Text Engineering Platform", Proceedings of the RANLP'2003 Conference, Borovets, Bulgaria, September 10-12, 2003.

[4] Berners-Lee, T., Hendler, J. and Lassila, O., The Semantic Web, *Scientific American*, May 2001. See http://www.scientificamerican.com/2001/0501issue/0501bernerslee.html

[5] C. Brewster, F. Ciravegna and Y. Wilks: User-Centred Ontology Learning for Knowledge Management, In Proceedings of 7th International Conference on Applications of Natural Language to Information Systems, Stockholm, June 27-28, 2002, Lecture Notes in Computer Science 2553, Springer Verlag.

[6] Handschuh, S., Staab, S., and Ciravegna, F, S-CREAM – Semi Automatic Creation of Metadata, Semantic Authoring, Annotation and Markup Workshop, 15th European Conf. on Artificial Intelligence, pages 27--33, Lyon, France, 2002.

[7] Vargas-Vera M., Motta E., Domingue J., Lanzoni M., Stutt A., Ciravegna F., MnM: Ontology Driven Semi-Automatic Support for Semantic Markup, EKAW 2002

[8] Vargas-Vera M., Motta E., Domingue J.,Shum B. S., Lanzoni M., Knowledge Extraction by using an Ontology-Based Annotation Tool, K-CAP 2001

[9] Alani H., Kim S., Millard D. E., Weal M. J., Web based Knowledge Extraction and Consolidation for Automatic Ontology Instantiation,

[10] Popov B., Kiryakov A., KirilovA., Manov D., Ognyanoff D., Goranov M., KIM – Semantic Annotation Platform,

[11] Kahan J., Koivunen M., Prud'Hommeaux E., Swick R.. Annotea: An Open RDF Infrastructure for Shared Web Annotations. In The WWW10 Conference, Hong Kong, May, pp. 623-632.

[12] Valarakos A., Sigletos G., Karkaletsis V., Paliouras G., A Methodology for Semantically Annotating a Corpus Using a Domain Ontology and Machine Learning, In RANLP, 2003

[13] Valarakos A., Paliouras G., Karkaletsis V., Vouros G., A Name Matching Algorithm for Supporting Ontology Enrichment, to appear In Proceedings of SETN'04

[14] N. F. Noy, R. W. Fergerson, & M. A. Musen, The knowledge model of Protege-2000: Combining interoperability and flexibility, In Proceedings of EKAW 2000, Juan-les-Pins, France, 2000

[15] Pazienza M.T., A. Stellato, M. Vindigni. "Combining ontological knowledge and wrapper induction techniques into an e-retail system", Proceedings of the International Workshop on Adaptive Text Extraction and Mining ECML/PKDD-2003, Cavtat-Dubrovnik, Croatia, September 22, 2003

[16] M. Volk, B. Ripplinger, S. Vintar, P. Buitelaar, D. Raileanu, B. Sacaleanu, Semantic Annotation for Concept-Based Cross-Language Medical Information Retrieval, In: International Journal of Medical Informatics, Volume 67:1-3, December 2002.

[17] Cohen, W., Ravikumar, P., Fienberg, S., A Comparison of String Distance Metrics for Name-Matching Tasks, In Proceedings of IIWeb Workshop, 2003

[18] Bontcheva, K., Dimitrov, M., Maynard, D., Tablan, V., Cunningham, H., Shallow Methods for Named Entity Co-reference Resolution, In Proceedings of TALN 2002, Nancy, 24-27 June 2002

[19] H. Alani, S. Kim, D.E.Millard, M.J. Weal, W. Hall, P.H.Lewis, N.R. Shadbolt, Automatic Ontology-Based Knowledge Extraction from Web Documents, IEEE Intelligent Systems, 2003