OILSW: A New System for Ontology Instance Learning in Semantic Web

Sima Soltani and Ahmad Abdollahzadeh Barforoush

Intelligent Systems Laboratory
Department of Computer Engineering and Information
Technology, Amirkabir University of Technology, Tehran, Iran
{ssoltani,ahmad}@ce.aut.ac.ir

Abstract. The Semantic Web is expected to extend the current Web by providing structured content via the addition of annotations. Because of the large amount of pages in the Web, manual annotation is very time consuming. Finding an automatic or semiautomatic method to change the current Web to the Semantic Web is very helpful. In a specific domain, Web pages are the instances of that domain ontology. So we need semiautomatic tools to find these instances and fill their attributes. In this article, we propose a new system named OILSW for instance learning of an ontology from Web pages of Websites in a common domain. This system is the first comprehensive system for automatically populating the ontology for websites. By using this system, any Website in a certain domain can be automatically annotated.

Keywords: Semantic Web, ontology, instance learning

1 Introduction

Nowadays, the Web is rapidly growing and becoming a huge repository of information, with several billion pages. Indeed, it is considered as one of the most significant means for gathering, sharing, and distributing information and services. At the same time this information volume causes many problems that relate to the increasing difficulty of finding, organizing, accessing and maintaining the required information by users. Recently, the area of the Semantic Web is coming to add a layer of intelligence to the applications that perform these processes.

As defined in (Berners-Lee, Hendler &Lassila, 2001) “the Semantic Web is an extension of the current Web in which information is given well defined meaning, better enabling computers and people to work in cooperation.” W3C has a more formal definition that “The Semantic Web is the representation of data on the World Wide Web. It is a collaborative effort led by W3C with

ARD Prasad & Devika P. Madalli (Eds.): ICSD-2007, pp. 54-63, 2007

participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework (RDF), which integrates a variety of applications using XML for syntax and URIs for naming" (W3C, 2001)

Semantic Web provides a common framework that allows data to be shared and reused in application, enterprise and community boundaries. A prerequisite for the Semantic Web is the availability of structured knowledge, so methods and tools need to be employed to generate it from existing unstructured content. Because the size of the Web is too huge and the manual annotation of Web pages is too time and resource consuming, an automatic or semiautomatic tool or method to transform the current Web content to the Semantic Web content is very useful in the process of developing Semantic Websites.

Ontologies are important components of Semantic Web, because they allow the description of semantics of Web content. The work of changing the Web to Semantic Web could be done by methods and tools of ontology learning and automatic annotation tools.

This paper is organized as follows. In section 2, we will define and describe ontology and ontology learning and their usage. In section 3, we will introduce our instance learning system, OILSW (Ontology Instance Learner for Semantic Web). Our experience on different classification algorithms on Web pages is discussed in section 4. In section 5, we will briefly review the related works in instance population and conclude in section 6.

2 Ontology Learning

Ontology is a philosophical discipline, a branch of philosophy that deals with the nature and the organization of being (Maedche, 2002). Ontologies are used for organizing knowledge in a structured way in many areas. We usually refer to an ontology as a graph/ network structure consisting from concepts, relations and instances. In a formal way, ontology can be defined as:

Definition: An ontology is defined by a tuple
$O = (C, T, \leq C, \leq T, R, A, \sigma_R, \sigma_A, \leq R, \leq A)$ which consists:

C as a set of concepts aligned in a hierarchy $\leq C$, R as a set of relations R with $\leq R$, the signature $\sigma_R : R \to C^2$, a set of data type T with $\leq T$, a set of attributes A with $\leq A$, and signature $\sigma_A : A \to C \times T$. For a relation $r \in R$ the domain and range of ontology define as $dom(r) := \Pi_1(\sigma_R(r))$ and $range(r) := \Pi_2(\sigma_R(r))$. (Ehrig, Haasa, Hefke & Stojanovic, 2005)

Today a completely automatic construction of good quality ontologies is in general impossible for theoretical, as well as practical reasons (Davies, Studer & Warren, 2006).

Depending on the different assumptions regarding the provided input data, ontology learning can be addressed via different tasks: learning just the ontology concepts, learning just the ontology relationship between the existing concepts, learning both concepts and relations at the same time, populating an existing ontology and other tasks. In (Davies, Studer & Warren, 2006) the ontology learning tasks defined in terms of mapping between ontology components where some of the components are given and some of them are missing, and with these tasks the missing ones induced. Some typical scenarios for these tasks are:

1) Inducing concepts /clustering of instances (given instances)
2) Inducing relations (giving concepts and the associated instances)
3) Ontology population (giving an ontology and relevant, but no associated instances.)
4) Ontology generation (given instances and any other background information)
5) Ontology updating/extending (given an ontology and background information, such as new instances or the ontology usage patterns)

Creating Semantic web from the current Web can be done in two ways: annotating the Web pages according to the domain ontology of that pages and constructing the web pages according to the domain ontology. In constructing Semantic Web pages, we can have a specified structure for each page and pages constructed according to that structure and all definitions in domain ontology.

Nowadays we have a large amount of Web pages, and if we want to annotate them manually or construct them again, it takes a lots of time and many mistakes may occurs. One of the efficient ways to do that is to annotate these pages automatically according to the domain ontology of those Web pages. This work is the scenario 3 which we have the domain ontology (concepts and relations) and we want to populate the ontology. For this work we should do some steps which we define in the next part.

3 OILSW: Our proposed ontology instance learner for Semantic Web

As we mentioned before, ontology instance learning or instance population is one of the important issues in ontology learning. Most of the works on ontology population have been done on information extraction and finding relation between pages in Web. But another problem in Web and in changing the current Web to the Semantic Web is for Website designers. It is hard and time consuming and also with lots of mistakes for designers to manually annotate their Websites. The number of pages in medium and large Websites occasionally is more than 1000 pages and this makes reconstruction of these Websites difficult. We propose a new system, named OILSW (Ontology

Instance Learner for Semantic Web) to do this task automatically in Websites with a common ontology (Figure 1). This system is a general system which different technologies can use in it's component.

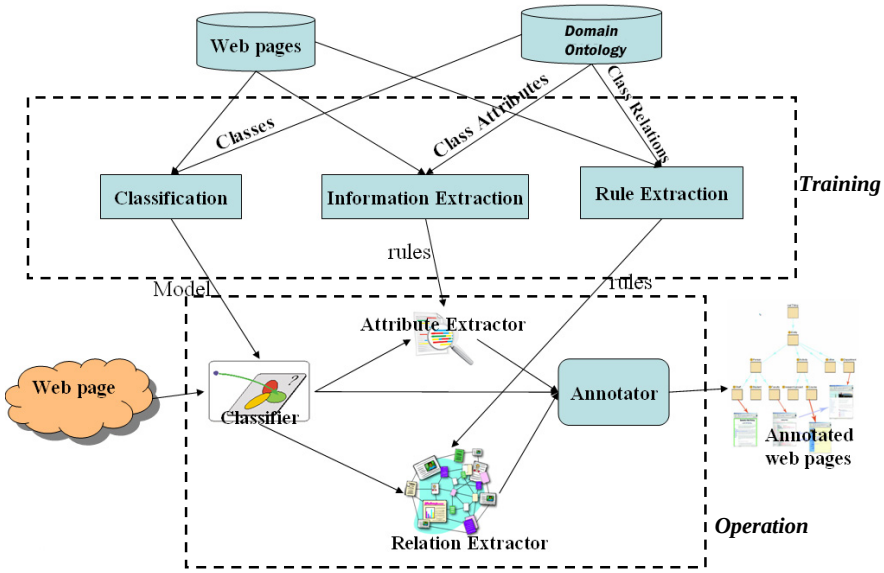

Figure 1: Architecture of OILSW

As shown in Figure 1, as an input of such a system, we have an ontology that belongs to the domain and Web pages of that domain (Web pages of Website) which we want to use as instances of that domain.

The system works on ontologies, which have an organization view. This means that this system works on websites that have this organizational structure, such as university web sites or companies websites. According to these ontologies and websites, each page in the website is an instance of the ontology class.

This system operates in two stages: *training* and *operation*. In *training* the system learns the rules and models and in *operation* the system uses these rules and models for annotating the Web pages. Training stage of system operation consists of the following steps:

1) Learning the classifier: Using classification algorithms
2) Learning rules of information extraction of that domain: using information extraction algorithms
3) Learning rules of relation extraction of that domain: using methods of extraction relations

For learning the classifier we have the ontology classes as classes of classifier and Web pages of that domain for training and testing the classifier. We implement this part as the most important part of the system and will describe in chapter 4. For information extraction attributes of ontology and Web pages of domain are inputs and for extracting the relations the relations in ontology and Web pages are the inputs

For information extraction component, we can use traditional IE or tools like Amilcare (Ciravegna & Wilks, 2003) and MnM (Motta , Vargas-vera & Domingue, Lanzoni, Stutt & Cirvegna , 2002) that extract the rules of information extraction according to the ontology with some revisions. In rule extraction also we can use the link and hyperlink properties or traditional IE to extract the rules which the rule extractor in the system can use for finding the relation between the instances.

The operation stage has 4 main tasks: *classification*, *information extraction*, *relation extraction* and *annotation.* Steps of operation are:

1) Relating each page to a class of the ontology (*Classifier*)
2) Finding the attributes of each instance (*Attribute extractor*)
3) Detect the relations between the instances that are the relations between the classes in ontology (*Relation extractor*)
4) Annotating the Web pages (*Annotator*)

Inputs and outputs of each step are as follow: Inputs of classifier are ontology classes, Web pages and classifier model and the output is the pages classified in ontology classes.

The first step in operation use classification models we have learned in the training. In attribute extractor we can use the information extraction rules we have learned in training to extract the proper values for attributes. The input of this step is information extraction rules, and the classified pages and the output is the attributes. For relation extractor we have learned rules for relation extraction in pages and classified pages that need to find their relations. This step can be done parallel with the second step of operation. The last step uses the output of previous steps and automatically annotates each page. With all these steps and components, and with domain ontology and training pages for that domain, the system can automatically learn the ontology instances on the web site and annotate them.

In all these four steps, the classifier could have a very distinct effect on the result of learning. Because its output is the input of other phases and any improvement in this phase could effect on the result of system distinctly. So using classification algorithms with better accuracy could help us in better results in the system. As the most important part of OILSW, we implemented the classification part and training the classifier model with 3 classification algorithms for Web pages. These are algorithms, which are used in text

mining. We have used the same to consider how proper they are when applied to web page classification. Our work was on the WEB-KB as a data set of Web to check what algorithms and feature selections can improve the result of this part.

4 Comparing Classification algorithms for classifying Web pages

Classification aids in better information retrieval and knowledge utilization but the sheer size of the Web along with the diversity of the subject matter makes manual classification a tedious and sometimes impractical task. It is highly desirable to be able to perform classification automatically employing computational techniques. Automated categorization is a supervised learning task in which new documents are assigned category labels based on the likelihood suggested by a training set of labeled documents.

For comparing classification on these domain specific data, we use the 3 most important & effective algorithms in text classification. These are Naïve Bays, K-Nearest Neighborhood and SVM. Naive Bays is a wildly used statistical learner known for its simplicity (Agrawal & Srikant, 2001) (McCallum & Nigam, 1998) (Chai, Ng & Chieu 2002). Support Vector Machines are one of the most accurate classifiers of text currently available. And KNN which is used in most of classification works (Yang, Zhang & Keisel, 2003) (Liu, Li & Wang 2002).

We use the WEB-KB dataset for our experience. In this dataset, we have 7 classes of pages: *course*, *Department*, *Faculty*, *Project*, *Staff*, *Student* and *Other* class which all pages that do not belong to the first 6 class will go to this class.

For classifying of Web pages, first of all we should tokenize them. In our tokenizer, we first detect all the tags, eliminate them and make the pages as a pure text. We tokenize the pages by selecting the words between separator marks (like space , commas, quotes and all others) as tokens. After that we tokenize all the pages in each class (according to the train set) and calculate the frequency of each word in that class after checking them with StopWords and stem them. So we eliminate the words in which are StopWords and can not have effect on classifying these pages. According to the (Craven, DiPasque, Freitag, McCallum, Mitchell, Nigam &Slattery, 2000) we delete some StopWords from the StopWords list, because they could use as features for classification (for example “I” is a word used in student’s pages for several times and it can show this class pages). After calculating the frequency of each word in these classes, we need a feature selection method to reduce the number of features for classification. According to (Yang & Pedersen 1997) document frequency (DF) is a reliable measure for selecting information features and it can be used instead of information gain (IG) or $\chi^2 - test$ (CHI)

when the computation of these measures is too expensive. So we use feature selection but another problem is that how many of the words should be in features. We check this by selecting some thresholds for document frequency of words. By changing the range of threshold we want to find the best features.

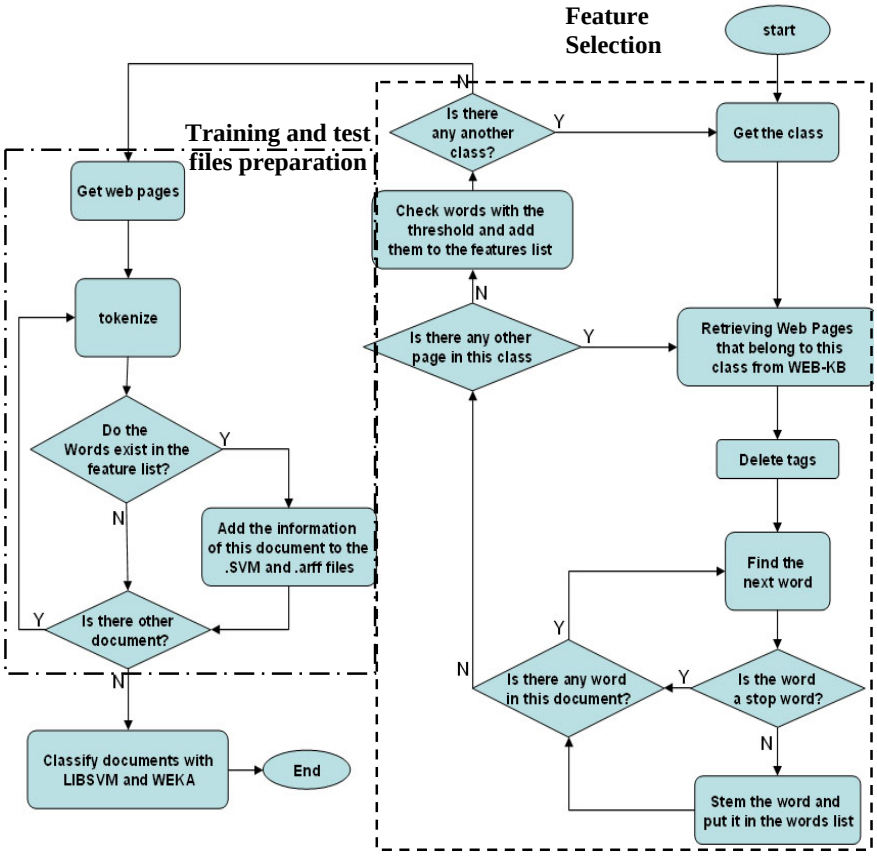

Figure 2. Feature Selection and Classification algorithm

After selecting each threshold we classified the documents by Naive Bays algorithm. The results have shown in Figure 3. As you can see in the figure, the threshold 30% (words that occurs in more that 30% of documents), have the best accuracy in the classification. After selecting the best threshold, system tokenizes the whole documents to find out the number of times the words repeated in them and then the system write these information in files. After that we use these files as inputs of classification tools (Figure 2).

For NB and KNN, we use WEKA, and for the SVM we use the LIBSVM as tools for classification. The results of our experience are shown in Table 1.

Table 1 shows accuracy of different classification algorithms on WEB-KB dataset.

Classification Algorithm	**NB**	**KNN**	**SVM**
Accuracy	31.775	76.7829	74.5642

Table 1. Results of different classification algorithms on Web data of a specific domain

As you can see, because the classes of this domain have common attributes, the accuracy is not high. For example 3 of our classes are subclasses of person class (*Student, Faculty and Staff*) and the *other* class has a combination of the other 6 classes. This makes more mistakes in classification of these for the algorithm. For this dataset, KNN has better accuracy than SVM and NB.

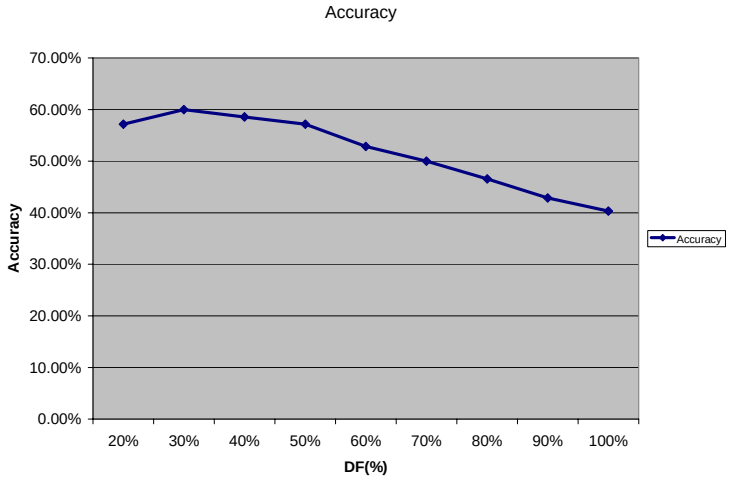

Figure3. Accuracy for different thresholds of DF

5 Related Works

(Geleijinse & Korst, 2005) Populate an ontology by the use of hand-crafted domain-specific relation patterns. Their algorithm uses instances of some class returned by Google to find instances of other classes. Work on information extraction from large Web sites, so called wrapper algorithms, can be found in (Crescenzi & Mecca, 2004). Brin (Brin, 1998) combines techniques from pattern extraction and wrapper algorithms to extract instances of relations from Web sites. Use of Google for identifying relation patterns can be found in (Cimiano & Staab, 2004).

6 Conclusion and Future Works

In this article, we discussed that one of the main challenges in developing the content for Semantic Web is transforming the existing Web content to

Semantic Web content by annotating the pages. We proposed a system for changing an existing Website to a Semantic Web Website by using ontology and ontology population on the domain of that Websites. This system helps the designers of Websites with common domain to annotate their websites automatically. Our system operates in training and operation stages. In training, the system learns the models and rules of classifiers, attribute extractors and rule extractor. In operation stage, the system uses these rules and models to annotate Web pages. We introduce the classification as the most important step in annotation, because its output is the input of the other 3 steps.

In the last part of our work, we had an experience in classification and we compared the 3 most common classification algorithms for classifying our Web pages and our experience showed that the KNN algorithm has the best accuracy on our data. The accuracy of all 3 was low because of the common features that the classes in our dataset have. The result of this experience shows that we need more effective algorithm for classifiers that work in Web domain.

For future works we propose implementing this system and work on different parts of this system for improving the algorithms that can be used in different parts, such as classification, information extraction and rule extraction to make better results in each part and as a result of the whole system.

References

[1] Agrawal, R. and Srikant, R. (2001). On integrating catalogs. *In Proceedings of the tenth international conference on World Wide Web, ACM Press* 603-612.

[2] Berners-Lee, T.; Hendler, J. and Lassilo, O., (2001), The Semantic Web, *Scientific American*, 284, 34-43.

[3] Brin, S. (1998). Extracting patterns and relations from the world wide Web, *In WebDB Workshop at 6th International Conference on Extending Database Technology*.172—183.

[4] Chai, M.K.; Ng, H.T. and Chieu, H.L. (2002). Bayesian online classifiers for text classification and filtering. *In Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval*, 97-104.

[5] Cimiano, P. and Staab, S. (2004). Learning by googling. *SIGKDD Explorations*, 6, 24-34.

[6] Ciravegna, F. and Wilks Y., (2003). Designing adaptive information extraction for the semantic web in Amilcare. In Handschuh S, Staab S, (eds) *Annotation for the semantic web*, ISO press.

[7] Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K. and Slattery, S. (2000). Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118, 69-113.

[8] Crescenzi, V. and Mecca, G. (2004). Automatic information extraction from large Websites. *Journal of the ACM*, 51, 731-779.

[9] Davis, John; Studer, Rudi and Warren, Paul (2006). *Semantic Web technologies: trends and research in ontology-based systems*. West Sussex: Wiley Publishing.

[10] Ehrig, M.; Haase, P.; Hefke M. and Stojanovic, N. (2005), Similarity for Ontologies - A Comprehensive Framework. *In 13th European Conference on Information Systems*

[11] Geleijnse G. and Korst J. (2005). Automatic ontology population by googling. *In Proceedings of the Seventeenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2005)*, 120 – 126.

[12] Liu, H.; Li, J. and Wong, L. (2002), A comparative study on feature selection and classification methods using gene and protein expression pro les, *in `Genome Informatics 2002: Proceedings of 13th International Conference on Genome Informatics', Universal Academy Press*, 51 - 60.

[13] Maedche, Alexander D. (2002). *Ontology Learning for the Semantic Web*. Kluwer Academic Publishers.

[14] McCallum, A., Nigam, K. (1998). A comparison of event models for Naive Bayes text classification. *In Learning for Text Categorization: Papers from the AAAI Workshop, AAAI Press*, 41-48.

[15] Motta E.; Vargas-vera M.; Domingue J.; Lanzoni M.; Stutt A. and Cirvegna F. (2002). MnM: Ontology deriven semi-automatic support for semantic markup. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pp: 379-391.

[16] W3C. http://www.w3.org/2001/sw

[17] Yang, Y. and Pedersen, J.O. (1997). A comparative study on feature selection in text categorization. *In Proc. of the Fourteenth International Conference of Machine Learning*, 412-420.

[18] Yang, Y.; Zhang, J. and Keseil, B. (2003), A scalability analysis of classifiers in text categorization. *In Proceedings of SIGIR-03, 26th ACM International Conference on Research and Development in Information Retrieval, ACM Press*.