A Survey of Ontology Learning Approaches

Maryam Hazman
Central Lab for Agricultural Experts Systems,
Ministry of Agriculture and Land Reclamation
Giza, Egypt

Samhaa R. El-Beltagy
Faculty of Computers and Information,
Cairo University
Giza, Egypt

Ahmed Rafea
Computer Science Department
American University in Cairo
Cairo, Egypt

ABSTRACT
The problem that ontology learning deals with is the knowledge acquisition bottleneck, that is to say the difficulty to actually model the knowledge relevant to the domain of interest. Ontologies are the vehicle by which we can model and share the knowledge among various applications in a specific domain. So many research developed several ontology learning approaches and systems. In this paper, we present a survey for the different approaches in ontology learning from semi-structured and unstructured date

General Terms
Ontology learning approaches.

Keywords
Ontology learning, Ontology learning evaluation, knowledge discovery.

1. INTRODUCTION
The World Wide Web is a vast and growing source of information and services which need to be shared by people and applications. Ontologies play a major role in supporting the information exchange and sharing by extending syntactic interoperability of the Web to semantic interoperability. Ontologies provide a shared and a common understanding of a domain that can be communicated between people and heterogeneous and distributed systems [1]. Also, semantic web and its applications rely heavily on formal ontologies to structure data for comprehensive and transportable machine understanding. Thus, the Semantic Web's success is dependent on the quality of its underline ontologies [2]. For reaching the goal of a semantic web, web resources need to be annotated with semantic information. Each of the users needs its appropriate ontologies that provide the basic semantic tools to construct the semantic web. Building such ontologies is not a new problem, knowledge engineers faces it in acquiring knowledge to develop knowledge-based systems.

Ontology can be regarded as a vocabulary of terms and relationships between those terms in a given domain. Examples of ontologies are WorldNet ontology [3], AGROVOC [4] and others. In other words, ontologies are meta-data schemas, providing a controlled vocabulary of concepts, each with an explicitly defined and machine process-able semantics. By defining shared and common domain theories, ontologies help both people and machines to communicate and support the exchange of semantics and not only syntax. The cheap and fast construction of domain specific ontologies is essential for the success and the proliferation of the Semantic Web [2]. The knowledge captured in ontologies can be used to annotate web pages, specialize or generalize concepts, drive intelligent search engine by using the relation between concepts existing in ontology.

In practical terms, an ontology may be defined as O = (C, R, A, Top), in which C is the non-empty set of concepts, R is the set of all assertions in which two or more concepts are related to each other, A is the set of axioms and Top is the highest-level concept in the hierarchy. R itself is partitioned to two subsets, H and N. H is the set of all assertions in which the relation is a taxonomic relation and N is the set of all assertions in which the relation is a non-taxonomic relation. There may also be bidirectional functions that relate the members of C and their motivating elements in the real world [5].

The remainder of this paper is organized as follows. In section 2 a brief description for ontology learning is presented. The unstructured and semi-structured ontology learning approaches will be discussed in sections 3 and 4. Section 5 introduces the methods for evaluating the ontologies built automatically or semi-automatically. Finally, concluding remarks are provided in section 6.

2. ONTOLOGY LEARNING APPROACHES
Manual acquisition of ontologies is a tedious and cumbersome task. It requires an extended knowledge of a domain and in most cases the result could be incomplete or inaccurate. Manually built ontologies are expensive, tedious, error-prone, biased towards their developer, inflexible and specific to the purpose that motivated their construction. [2] [6] [5] [7]

Researchers try to overcome these disadvantages of manual building ontology by using semi-automatic or automatic methods for building the ontology. Automation of ontology construction not only reduces costs, but also results in an ontology that better matches its application [7]. During the last decade, several ontology learning approaches and systems have been proposed. They try to build ontology by two ways. One way is developing tools that are used by knowledge engineering or domain experts to build the ontology like Protege-2000 [8] and ontoEdit [9]. Another way is semi-automatic or automatic building the ontology by learning it from different information sources [6] [5].

Ontology learning refers to extracting ontological elements (conceptual knowledge) from input and building ontology from them. [5]. It aims at semi-automatically or automatically building ontologies from a given text corpus with a limited human exert. Ontology learning can be defined as the set of methods and techniques used for building ontology from scratch, enriching, or adapting an existing ontology in a semi-automatic fashion using several sources [6]. Ontology learning

uses methods from a diverse spectrum of fields such as machine learning, knowledge acquisition, natural-language processing, information retrieval, artificial intelligence, reasoning and database management [7] [6].

Ontology learning systems can be categorized according to the types of the date from which they are learned [6] [5]. These types of data are unstructured, semi-structured, and structured. Unstructured data is the natural text like books, journals. Semi-structure date is text in HTML, XML files. While structured date are the databases and dictionaries. We will concentrate on ontology learning from unstructured and semi-structured types in this survey.

3. LEARNING FROM UNSTRUCTURED DATA

Unstructured data is the most difficult type to learn from. It needs more processing than the semi-structure data. The systems which have been proposed for learning from free text, often depend on natural language processors. Some systems used shallow text processing with statistical analysis like [10] and others use a rule based parser to identify dependency relations between words in natural language Sabou et.al. [7]. Cimiano et. al. [11] use the part of speech tagger TreeTagger [12] and the parser, LoPar2 [13]. Cimiano and Vaolker [14] extract ontologies from natural language text using statistical approach, pattern matching approach and a machine learning approach with the basic linguistic processing provided by Text2onto.

In our survey we found out that NLP is common among all techniques. Therefore, we classify the different approaches based on the technique used in addition to NLP. The first section describes a system which is an example of integrating NLP with statistical approach that uses the frequency count of noun and noun phrases in documents retrieved from the web to discover concepts and taxonomical relations while using shallow parser to extract noun phrases. The second section describes a pure NLP system which uses dependency grammar and parsers to discover the relation between syntactic entities. The third section describes an integrated approach that includes methods from different disciplines namely: Information Retrieval, Lexical database (WordNet), machine learning in addition to computational linguistics.

3.1 Statistical Approach

Sanchez and Moreno [10] start building ontology using keywords that are near to ontology concepts and closely related. They send initial keyword's to search engine for retrieving the related pages, then analyze these web sites in order to find important candidate concepts for a domain. This keyword is used for learning its children concepts from the returned pages by retrieving the bigrams that contain the keyword as the second term. For example if the keyword is biosensor and the immediate anterior word is optical (e.g. optical biosensor) then optical biosensor is a candidate children concept for biosensor if it have a minimum size and is not a stop words. Selecting the representative concepts from the candidate concepts according to the following attributes:-

- Total number of appearances (on all the analyzed web sites)
- Number of different web sites that contain the concept
- Estimated number of results returned by the search engine setting the selected anterior word alone (e.g. optical).
- Estimated number of results returned by the search engine joining the selected concept with the initial keyword.
- Ratio between the two last measures.

Only candidate concepts whose attributes fit with a set of specified constraints (which is a range of values for each parameter) are selected. This system uses stemmed terms while counting the number of occurrence of the terms to enhance its performance in discovering concepts. They consider these discovered concepts as new keywords and rerun their system again to discover their children concepts. This process is repeated recursively until a selected depth level is achieved or no more results are found. The obtained result is a hierarchy that is stored as ontology.

3.2 Natural Language Processing Approach

Sabou et.al. [7] use a set of syntactic patterns to discover the dependency relations between words. Their extraction method exploits the syntactic regularities which are inherent from the sublanguage nature of web service documentations, which is a specialized form of natural language. Their ontology extraction steps are: dependency parsing, syntactic patters, ontology building, and ontology pruning. They use a dependency parsing to identify dependency relations between words in natural language. A dependency relation is an asymmetric binary relation between a word called head and a word called modifier. For example, in the sentence “Find antigenic sites in proteins” the “antigenic” is an adjective which modifies the noun “sites”, and “sites” is the object of the verb ‘find”. Then, a set of syntactic patterns is used to identify and extract interesting information from the annotated corpus for ontology building.

They define three major group/categories of patterns used to derive different types of information. First group is used for identifying domain concepts. Here, the noun and noun phrase patterns ("NN” and “NMod") are used for discovering concepts and dependency relations between them (like, <antigenic site> and <site>). Second group is used for identifying functionalities that are frequently offered in that domain using verbs to identify the functionality performed by a method and nouns closely related to these verbs (like, <find> <antigenic site>). The last groups are used for identifying relations using the prepositional phrases (PP) to identify a meronymy relation between the terms that they interrelate (like, find antigenic sites in proteins “in proteins” is the PP <antigenic sites> are parts of a <protein>).

Cimiano et. al. [11] present an automatic approach for acquiring taxonomies or concept hierarchies from a textual corpus. Their approach is based on Formal Concept Analysis which discovers inherent relationships between objects described through a set of attributes and the attributes themselves [15].

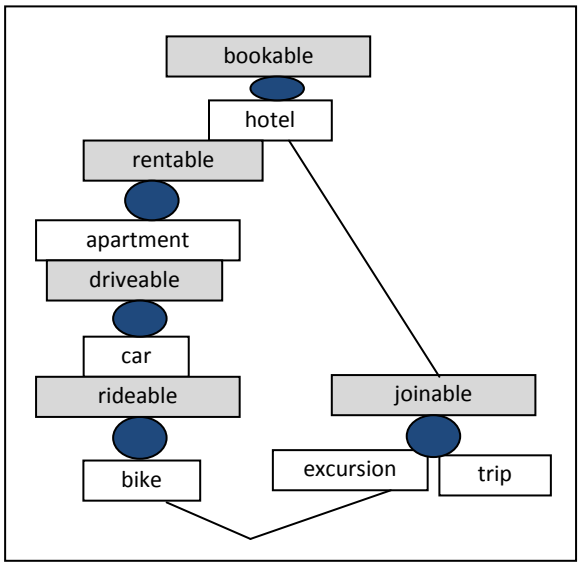

Fig 1: The lattice of formal concepts for the tourism example (ref: [11])

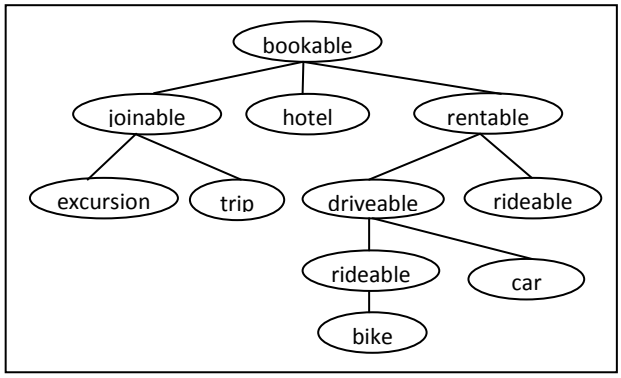

Fig 2: the corresponding hierarchy of ontological concepts for the tourism example (ref: [11])

First, they parse the corpus to tag its words by their part-of-speech and generate parse trees for each sentence. The verb/subject, verb/object and verb/prepositional phrase dependencies are extracted from these parse trees. Then, the verb and the heads are lemmatized. As the assumption of completeness of information will never be fulfilled, the collection of pairs is smoothed. The smoothing is done by clustering all the terms which are mutually similar with regard to the similarity measure in question. Counting more attribute/object pairs than are actually found in the text will lead to obtaining non-zero frequencies for some attribute/object pairs that do not appear literally in the corpus. The overall result is thus a 'smoothing' of the relative frequency landscape by assigning some non-zero relative frequencies to combinations of verbs and objects which were actually not found in the corpus. For example, car and bike are mutually similar, and consequently the pairs having any of them with their verb attributes, will be clustered together. The object/attribute pairs are weighted using conditional probability, point wise mutual information and the relative entropy of the prior and posterior distributions of a set of pairs to determine 'selectional strength' of the verb at a given argument position. Only pairs over a certain threshold are transformed into a formal context to which Formal Concept Analysis is applied to produce ontology in lattice form (figure 1). Formal Concept Analysis is a method based on order theory and used for the analysis of data, in particular for discovering inherent relationships between objects described through a set of attributes on the one hand, and the attributes themselves on the other [16]. Then the result is transformed from the lattice form to a partial order form which is closer to a concept hierarchy (figure 2).

3.3 Integrated Approach

Text2Onto [14] assists its users in selecting an appropriate learning algorithms for the kind of ontology they wants to learn. First, the corpus is parsed to annotate by part-of-speech and stemming its words. Text2onto have a library of algorithm to learn different ontology elements. These elements are concepts, concept inheritance, concept instances, general relations, metrological relations (part of), and equivalence.

Learning concepts algorithms depend on this approach is based on the assumption that a frequent term in a set of domain specific texts indicates occurrence of a relevant concept. So, they learn concepts using Relative Term Frequency (RTF), TFIDF (Term Frequency Inverted Document Frequency), Entropy and the C-value/NC-value method [16]. For extracting concept inheritance relations text2onto have implemented various algorithms depending on exploiting the hypernym structure of WordNet, matching Hearst patterns and applying linguistic heuristics rules. In order to learn general relations, Text2Onto employs a shallow parsing strategy to extract sub categorization frames enriched with information about the frequency of the terms appearing as arguments. In particular, it extracts the syntactic frames like, love(subj,obj) and maps this subcategorization frames to ontological relations. Mereological (Part_of) Relations is learned using patterns matching technique. Learning concept instances relations rely on a similarity-based approach extracting context vectors for instances and concepts from the text collection and assigning instances to the concept corresponding to the vector with the highest similarity. Also, they use a pattern-matching for learning concepts instances. Equivalence relations are learning following the assumption that concepts are equivalent to the extent to which they share similar syntactic contexts. After the process of ontology extraction is finished, the ontology is presented to the user for refining it. Finally, the user can select among various ontology writers, which are provided for translating the learned ontology into different ontology representation languages.

4. LEARNING FROM SEMI-STRUCTURED DATA

Building ontology from semi-structure data uses both traditional data mining and web content mining techniques. Karoui et. al [1] and Bennacer and Karoui [17] use the Web pages structure to build a database table then use clustering method to build their ontologies. They use the structure of the HTML file with some linguistic as features to identify the candidate concepts. While Davulcu et. al. [18] convert the html page to hierarchical semantic structures as XML to mine it for generating taxonomy. Hazman et.al. [19] build ontology through the use of two complementary approaches. The first approach utilizes the structure of phrases appearing in the documents' HTML headings while the second utilizes the hierarchical structure of

the HTML headings for identifying new concepts and their taxonomical relationships between seed concepts and between each other. The following subsections describe these two approaches namely: Data Mining and Web content mining.

4.1 Data Mining Approach

Karoui et. al. [1] use clustering techniques to group similar words into clusters in order to define a concept hierarchy. First, they exploit the text and HTML page structure to generate concepts. The HTML pages are processed to keep title, sub title, bold, italic, underlined, big character, keywords, hyperlinks, list, paragraph tags and the associated full text. They build a data table whose fields contain the word, the labeled word (concept to which the word belongs), the grammatical type of the word (noun, adjective, etc), the style of the word (title, bold, etc), a number representing how many times the word in this HTML tag style appears in the document and the number of documents that locate the word. They group words referring to the same meaning through user interaction. They use unsupervised method which is a divisive clustering [20] method to generate hierarchy of concepts clusters. A concepts cluster is described in terms of the words, it contains, and belonging to all the tag styles except the paragraphs tags or hyperlinks tags.

Also, Bennacer and Karoui [17] transform HTML web pages into structured data represented by a relational table (database). Then this relational representation is enriched by characterizing its structural and linguistic features in order to determine precisely the context of a term and its vicinity. The web pages are processed to keep only the text associated to a set of markups (such as <h1>, <b>, <i>, and <li>) considered to be important to retrieve the most important terms. To emphasize important terms, they define the <TITLE_URL> tag for hyperlink, <CHOICE> tag for a check box, <KEYWORDS> tag to all elements of Meta data associated to a document. The output of this step is represented in database table. The table attributes are term, its markup (associated tag), its previous associated tag (<h1> is a previous a tag for <h2>) and its ranking (They put a degree of the importance of these tag 1 for <title> and <h1>, 2 for <li> list items) in its source document are filed from this step. They use three kinds of analysis in order to evaluate and to characterize structural, nature and linguistic corpus features. Structure Analysis evaluates the structural features of the considered corpus by computing markup frequency for each markup category (tag <h1> category), and associated term percentage (museum and <h1>). Also, it discovers structural patterns to determine markups that appear together (<h1>-> <p>). These structural patterns allow the user to refine the term context definition by delimiting its vicinity.

Nature analysis analyses the HTML pages corpus selected to determine if changing the corpus content by removing or adding HTML documents until obtaining homogeneous covering the considered domain. Linguistic analysis and characterization identify the term stem and the syntactic category (verb, noun, adjective, adverb, etc.) of the stem. They use the TreeTagger tool [12] in order to assign a syntactic category and a stem to each term of the corpus. This information enriches the relational table by filling attributes related to linguistic characteristics. Also they derive patterns (term lemma, its grammatical type) which are used for refining the definition of term context and its semantic relation.

For clustering, they use a similarity or distance measure in order to compute the pair wise similarity or distance between vectors corresponding to two terms in order to decide if they can be clustered or not. The user can compare the results obtained by applying different similarity measures like (cosine, Euclidian distance, jaccard, etc). They combine co-occurrence in a structural context (using structure patterns) and co-occurrence in a syntactic context (using syntactic patterns) to weight the significance of a given term pairs. If two terms occur in the same block level tag (<h1> </h1>) the context is delimited by the tag and their co-occurrence is computed in this context. If two terms occurred in different tags that are related structurally (<h1>,<p>) their co-occurrence is computed regarding this link in this context. The initial hierarchy cluster is obtained from keywords tags corresponding to the most important terms. Leaf clusters are then refined by considering each co-occurrence terms in both structural and syntactic contexts. They build a tree to represent markup hierarchy to guide clustering procedure to iteratively consider two terms belonging to the considered hierarchy level. This iterative clustering allows the user to evaluate cluster at each step. After each iterative, the user exam and validate the clusters.

4.2 Web Content Mining Approach

Davulcu et. al. [18][21] developed OntoMiner which learns from html pages to build taxonomy using their structure only. OntoMiner is an automated techniques for bootstrapping and populating specialized domain ontologies by organizing and mining a set of relevant overlapping taxonomy-directed domain specific Web sites that provided by the user and characterizes her domain of interest. A taxonomy-directed web site is web site that contains at least one taxonomy for organizing its contents and presents the instances belonging to a concept in a regular fashion (like scientific, news, and travel). As shown in figure 3, Web pages are crawled and passed to the semantic partition module which partitions the Web page into logical segments and generates the Document Object Model (DOM) tree. Finally it uses promotion rules that are based on the presentation and the format of the Web page to promote the emphasized labels (e.g. the group of words appearing in a heading or in a bullet…) with tags like <b>, <U>, <h1>, on top of certain groups as its parent xml node. Taxonomy mining module first mines for frequent labels in the XML documents. The labels that have frequency more than the threshold are separated from the rest of the document as important labels (e.g., Business, Sports, Politics, Technology, Health, and Entertainment are important concepts in the News domain). For missed labels that are relevant but infrequent, they learn attributed tag paths of the frequent labels and then apply them within the corresponding logical segments to retrieve more labels. For example, they identified Entertainment to be a frequent label and it has the same tag path as Culture which is infrequent label. Also they use some rules to eliminate the irrelevant labels. For example they ignore a label if it does not have hyperlink. These important labels are stemmed, and organized into groups of equivalent labels (e.g. "Sport" and "Sports" are grouped together). Each collection of labels is considered as a concept c. These concepts are flat. Organizing these concepts into taxonomy required mining is-a relationship from the semantically partitioned Web pages (The child-parent relation in the XML tree). To expand the domain taxonomy, they follow the hyperlinks corresponding to every concept c. For example, sport is a concept, the pages that are hyperlinked by

the words corresponding to the concept "sport", will be used for building the sport sub-taxonomy) and expand the taxonomy depth-wise. Finally, they mine the concept instances (members of concepts) and the values of the instance attributes in the same way as the sub taxonomy mining.

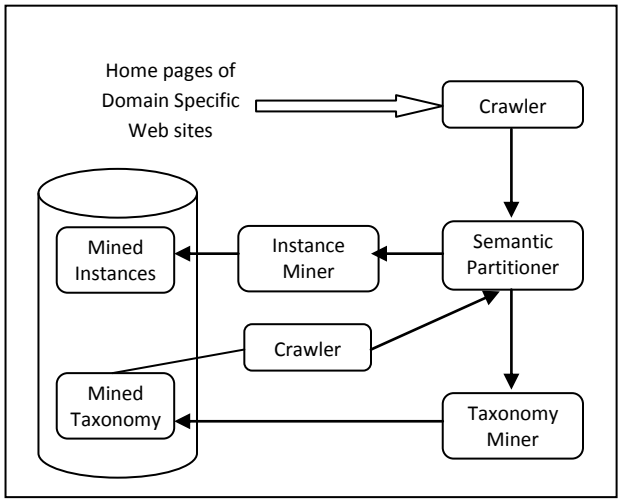

Fig 3: Architecture of OntoMiner (ref: [18][21])

Hazman et.al. [19] use both the structure of phrases appearing in the documents' HTML headings and the hierarchical structure of the HTML headings for identifying new concepts and their taxonomical relationships between seed concepts and between each other.

The architecture of their proposed system is given in figure 4. First, the heading extractor extracts headings from input HTML documents in order to enable their mining for the purpose of concept extraction. The extracted heading are normalizes by the Heading Preprocessor. It normalizes heading text by removing any numbers or stop words contained within it and by stemming it. Their first learning approach is the N-gram based Ontology learner. It extracts concepts and their taxonomical relation using word sequences (N-gram phrases) in text headings. It tries to find their children for the seeding concepts in the heading text by extracting all possible phrases (n-gram words) that have one of the seed concepts as their headword. Trying to locate the seed as a headword is specific to Arabic. For example, given the seed concept "disease", and a heading title of "powdery mildew disease", the n-gram learner would consider the phrase "powdery mildew disease" as well as the word "powdery mildew" candidate phrases.

The extracted ontology may include fake concepts, so they use a set of filters that can be applied to remove noisy or fake concepts. Sometimes the seed concepts are not act as a headword to their children concepts. So they used the heading structure of input Web documents to learn ontology in their second approach. In this approach the structure of the HTML document (heading levels) is used to learn the taxonomical ontology. They locate the seed concepts at the top level headings of the document set, consider the concepts at the second level as the children of the top level, and the concepts at the third level as the children of the second level, etc. The HTML Ontology Refiner is used to extend the ontology extracted by this approach. It discovers new concepts that have sibling relations with previously learnt concepts.

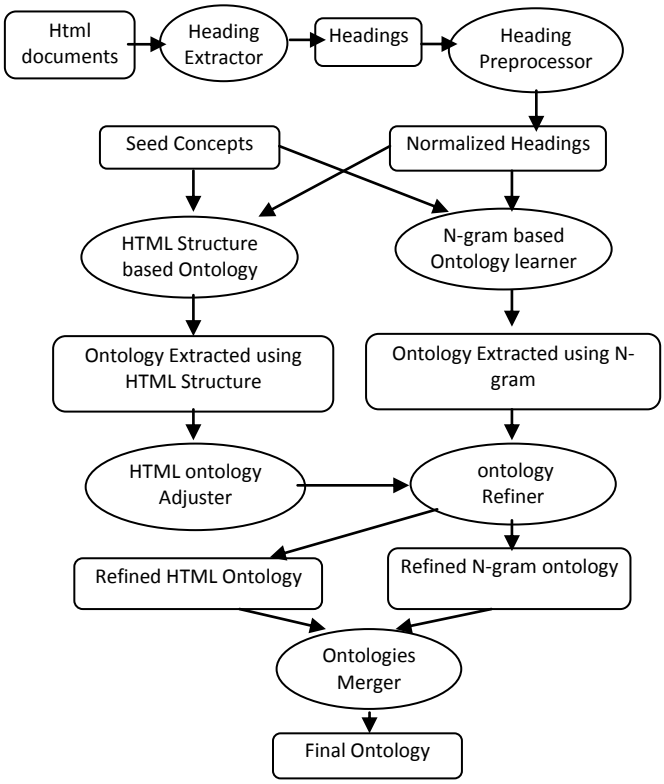

Fig 4: The ontology learning process (ref: [19])

Merging the constructed ontologies is done by the ontology Merger. This module takes both the N-gram based Ontology learner and the HTML structure based Ontology learner and merges them.

5. EVALUATION METHODS

It has been strongly argued that a key factor in making a particular discipline or approach scientific is the ability to evaluate and compare the ideas within the area. Evaluation, in general, means to judge technically the features of a product. It seems that having a trustworthy ontology information source is extremely important.

Ontologies are to be widely adopted in the semantic web and other semantics-aware applications so its evaluation becomes an important issue to be addresses. Users facing many of ontologies need to have a way of assessing them and deciding which one best fits their requirements. Also, people constructing ontology need a way to evaluate the resulting ontology. Intermediate evaluation can guide the construction process and any refinement steps. Automated or semi-automated ontology learning techniques require effective evaluation measures helping to select the "best" ontology out of many candidates [22]. There are two types of evaluation: ontology (content)

evaluation and ontology technology evaluation. Evaluating ontology is a must for avoiding applications from using inconsistent, incorrect, or redundant ontologies. A well evaluated ontology won't guarantee the absence of problems, but it will make its use safer. Evaluating ontology technology will ease its integration with other software environments, ensuring a correct technology transfer from the academic to the industrial world [23].

An ontology is a complex structured, so it is more practical to focus on the evaluation of different levels of the ontology separately rather than trying to directly evaluate the ontology as a whole. The broadly similar and usually involve are the following levels:

- Lexical, vocabulary (data layer), in which concepts, instances, facts, etc. have been included in the ontology, and the vocabulary used to represent or identify these concepts.
- Hierarchy (taxonomy), in which a hierarchical is-a relation between concepts is included in the ontology.
- Context (application level) when an ontology may be part of a larger collection of ontologies, and may reference or be referenced by various definitions in these other ontologies. In this case it may be important to take this context into account when evaluating it. Another form of context is the application where the ontology is to be used; evaluation looks at how the results of the application are affected by the use of the ontology.
- Syntactic level, evaluation on this level may be of particular interest for ontologies that have been mostly constructed manually.
- Structure, architecture, design, evaluation on this level use when wanting the ontology to meet certain pre-defined design principles or criteria; structural concerns involve the organization of the ontology and its suitability for further development [22].

Evaluated ontology approaches can be categorized to [22]:

- Gold Stander evaluation: Comparing the ontology to a "golden standard" like Sabou et.al. [24]. In a gold standard based ontology evaluation the quality of the ontology is expressed by its similarity to a manually built gold standard ontology. A "golden standard" is a predefined ontology is usually built manually from scratch by domain experts. One of the difficulties encountered by this approach is that comparing two ontologies is rather difficult. Measuring the similarity between ontologies can done by compare ontologies at two different levels: lexical and conceptual. [25].
- Application based evaluation: Using the ontology in an application and evaluating the results. This evaluation is used when an ontology is developed in order to be used in a specific application. The ontology is evaluated by use it in some kind of application or task. Then the evaluation of the outputs of this application, or its performance on the given task will be used as evaluation for the used ontology [22]. A system performs well if the query computation time is low, the reasoning is efficient enough, the answers are the correct ones and these ones that are produced are all that could be produced, etc.
- Data-driven evaluation: Comparisons with a source of data about the domain to be covered by the ontology [10]. These are usually collections of text documents, web pages or dictionaries. An important required for the data sources is to be representative and related to the problem domain to which the ontology refers. This kind of evaluation is preferable in order to determine if the ontology refers to a particular topic of interest.
- Human evaluation: Human evaluation is the most popular evaluation method. The evaluation is done by humans who try to assess how well the ontology meets a set of predefined criteria, standards, requirements, etc. [21]. It includes technical evaluation by the development team or by domain experts, and end users.

The four major categories of ontology evaluation aim at the assessment of ontologies in various layers. However they cannot deal with the evaluation of ontology as a whole. For example, data driven evaluation can be used to evaluate the lexical, hierarchical and the relational layer of ontology, but not the structural. While, a golden standard approach cannot evaluate the contextual layer. Human evaluation seems to be able to assess multiple ontological layers. Table 1 shows the relations between these approaches and ontology evaluation levels. There is no single best or preferred approach to ontology evaluation. The choice of a suitable approach must depend on the purpose of evaluation, the application in which the ontology is to be used, and on what aspect of the ontology that are being tried to evaluate [22].

Table 1. An overview of approaches to ontology evaluation (ref: [22])

Level	Approaches			
	Golden Standard	**Application Based**	**Data Driven**	**Human**
Lexical, vocabulary, data	X	X	X	X
Hierarchy, taxonomy	X	X	X	X
Semantic relations	X	X	X	X
Context, application		X		X
Syntactic	X			X
Structure, architecture, design				X

6. CONCLUSION

The problem that ontology learning deals with is the knowledge acquisition bottleneck, that is to say the difficulty to actually model the knowledge relevant to the domain of interest. Ontologies are the vehicle by which we can model and share the knowledge among various applications in a specific domain. Ontologies play a central role in the Semantic Web and can be used to enhance existing technologies from machine learning and information retrieval. So many research developed several ontology learning approaches and systems.

Their approaches have different features according to achieve their deferent goals. Some try to build concepts only [10] or concepts with their hierarchy [21] [7] [11] [1] [19]. Others build different types of ontology elements, like Text2Onto concerned by build concepts, concept hierarchy, concept instantiation, relations and equivalence terms [14].

According to their output their approached are vary between linguistic, heuristic and pattern matching (Logical), machine learning and statistical techniques. Statistical approaches are used to build ontology like frequency of the terms in [18], [10] and [1]. Also heuristic rules can be used in generate ontology [2]. Sabou et. al. used heuristic rules and linguistic-based [7]. Machine learning uses in building taxonomy by clustering the candidate terms relies on some similarity measures between the extracted terms in [17] [1]. Clearly, linguistic techniques for require Natural Language Processing (NLP) and they depend on tools for (POS) tagging, stemming, etc. and it used with other techniques like machine learning in [14]. Using linguistic techniques and pattern matching led the system to be a language dependent.

Some system start building ontology from scratch like [1] [18] [7]. While other can aim by some keywords that to be representative enough for a specific domain [19] [10]. Others import and reuse existing ontologies [2]. Also The ontology learning systems different in their degrree of automation from semi-automatic [1] [19], cooperative [2], fully automatic [10] [18] [7].

As observe evaluation the ontology is an important task, since ontology reflects in the performance of the application using it. Ontology evaluation is still remaining an important open problem.

7. REFERENCES

[1] Karoui, L., Aufaure, M., and Bennacer, N. 2004. Ontology Discovery from Web Pages: Application to Tourism. In ECML/PKDD 2004: Knowledge Discovery and Ontologies KDO-2004.

[2] Maedche, A. and Staab, S. 2001. Ontology Learning for the Semantic Web. In IEEE Intelligent Systems, Special Issue on the Semantic Web, 16(2).

[3] Fellbaum, C. 1999. Ed.: "WordNet: An Electronic Lexical Database", MIT Press.

[4] Soergel, D. Lauser, D., Liang, A., Fisseha, F., Keizer, J., and Katz, S. 2004. Reengineering Thesauri for New Applications: the AGROVOC Example. In Journal of Digital Information, 4, 4 (Mar. 2004).

[5] Shamsfard, M. and Barforoush, A. A. 2003. The state of the art in ontology learning: A framework for comparison. The Knowledge Engineering Review, Vol. 18 No.4 pp. 293-316.

[6] Gomez-Perez, A., Manzano-Macho, D. 2003. OntoWeb Deliverable 1.5: A Survey of Ontology Learning Methods and Techniques. Universidad Politecnica de Madrid.

[7] Sabou, M., Wroe, C., Goble, C., and Mishne, G. 2005. Learning Domain Ontologies for Web Service Descriptions: an Experiment in Bioinformatics. In Proceedings of the 14th International World Wide Web Conference (WWW2005), Chiba, Japan.

[8] Noy, N. F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R.W., and Musen, M.A. 2001. Creating Semantic Web Contents with Protege-2000. In IEEE Intelligent Systems, Vol. 16, No. 2, pp. 60-71.

[9] Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., and Wenke, D. 2002. OntoEdit: Collaborative ontology development for the semantic web. In International Semantic Web Conference 2002 (ISWC 2002), Sardinia, Ital.

[10] Sanchez, D., and Moreno, A. 2004. Creating ontologies from Web documents. In Recent Advances in Artificial Intelligence Research and Development. IOS Press, Vol. 113, pp.11-18.

[11] Cimiano, P., Hotho, A., Staab, S. 2005. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. JAIR - Journal of AI Research, Vol. 24, pp. 305-339.

[12] Schmid, H. 1994. Probabilistic part-of-speech tagging using decision trees. In Proceedings of the International Conference on New Methods in Language Processing.

[13] Schmid, H. 2000. Lopar: Design and implementation. In Arbeitspapiere des Sonder for schungsbereiches, No. 149.

[14] Cimiano P., and Vaolker, J. 2005. Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery. In: Montoyo, A., Munoz, R., Metais, E. Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB), Lecture Notes in Computer Science. Alicante, Spain: Springer.

[15] Ganter, B. and Wille, R. 1999. Formal Concept Analysis - Mathematical Foundations. Berlin:Springer-Verlag.

[16] Frantzi, K., Ananiadou, S., and Tsuji, J. 1998. The c-value/nc-value method of automatic recognition for multi-word terms. In Proceedings of the ECDL .pp 585-604.

[17] Bennacer, N., and Karoui L. 2005. A framework for retrieving conceptual knowledge from Web pages. In Semantic Web Applications and Perspectives, Proceedings of the 2nd Italian Semantic Web Workshop, University of Trento, Trento, Italy.

[18] Davulcu, H., Vadrevu, S., and Nagarajan, S. 2004. OntoMiner: Bootstrapping Ontologies From Overlapping Domain Specific Web Sites. In: Poster presentation at the

13th International World Wide Web Conference May 17-22 2004, New York, NY.

[19] Hazman, M., El-Beltagy, S. R., and Rafea, A. 2009. Ontology Learning from Domain Specific Web Documents. In International Journal of Metadata, Semantics and Ontologies, Vol. 4, No. 1-2, pp: 24 – 33.

[20] Touati, M., and Chavent, M. 2001. DIV: A divisive and symbolic clustering method, SFC2001, IXemes journees de la societe Francophone de Classification.

[21] Davulcu, H., Vadrevu, S., Nagarajan, S., and Ramakrishnan, I. 2003. OntoMiner: Bootstrapping and Populating Ontologies from Domain Specific Web Sites. In IEEE Intelligent Systems, Vol. 18, No. 5, pp. 24-33.

[22] Brank, J., Grobelnik, M., and Mladenic, D. 2005. A survey of ontology evaluation techniques. In Proceedings of the 8th Int. multi-conference Information Society IS-2005.

[23] Sure, Y., Daelemans, W., Perez, G.A., Guarino, N., Noy, N., and Reinberger, M. 2004. Why evaluate ontology technologies? Because they work!. In IEEE Intelligent Systems. Vol. 19, No. 4, pp. 74-81.

[24] Sabou, M., Lopez, V., Motta, E. and Uren, V. 2006. Ontology Selection: Ontology Evaluation on the Real Semantic Web, In: Proceedings of the 4th internacional EON workshop (EON'06) Maio.

[25] Dellschaft, K. and Steffen, S. 2006. On How to Perform a Gold Standard Based Evaluation of Ontology Learning. In: I. Cruz et al. The Semantic Web - ISWC 2006, Lecture Notes in Computer Science Vol. 4273/2006, pp228-241, Heidelberg, Germany: Springer Berlin