A Survey on Semantic Annotation Tools

Mariam Gawich
Faculty of Computing and Information Technology, Arab Academy, Cairo, Egypt.
mariamjawich@yahoo.fr

A.Badr [1], A.Hegazy[2], H.Ismael[2]
[1]Faculty of Computers and Information, Cairo University, Cairo, Egypt.
a.badr.fci@gmail.com
[2] Faculty of Computing and Information Technology, Arab Academy.
ahegazy@aast.edu,drhanafy@yahoo.com

Abstract

Ontology refinement is the process of adding new information to the webpage and to the ontology used in a specific domain. This process is applied through the use of semantic annotation tool. The ontology refinement helps to manage knowledge on web by indexing, retrieving and creating metadata which is accessible by autonomous agents. This paper presents a comparative analysis between Ontomat and GATE as tools for semantic annotation. The study needs for enhancement on the ontology extraction phase when it is applied in a biochemistry field.

Keywords: *Semantic Annotation, Semantic Web, Ontology, Ontology Refinement, Knowledge Management.*

1. Introduction

Using search engines ranks billion of web pages and identifying candidates, they often present the page a user wants within first few search. These engines are keyword based searching restricts the type of questions people can ask.Example: user make request like "find hair treatment for under 100$, it should contain biotin". The search engine can't answer this type of question because it doesn't know the relation between hair and biotin; it can't match the specified concepts. Semantic annotation attempts to solve this question.

Semantic annotation (SA) is the approach proposed within the framework of semantic web for creating such metadata [1].SA refers to the process of indexing and retrieving useful knowledge from documents by adding metadata to the documents content given agreed domain ontology .These metadata or annotations can be exploited by both humans, machines and make information accessible to autonomous agents[1].Semantic annotation is used to enhance search, information visualization and reasoning web resources.

In semantic web, ontology is defined as formal explicit specification of shared conceptualization [2] where formal implies that the ontology should be machine-readable and shared that it is accepted by a group or community. It is a method of knowledge representation.

The Figure below [Figure (1)] shows a typical semantic annotation tool receives an ontology domain as a file coded with xml or rdf-xml or owl or onto file. It uses an algorithm to import the ontology file, extract ontology, pruning ontology and refinement ontology. It brings out the webpage and the annotation which is compatible with web agents plus the updated ontology file.

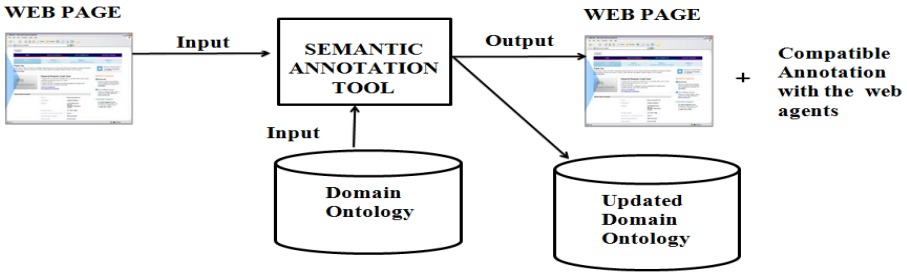

Figure (1): Semantic annotation tool

Semantic annotation tools can be classified manual that allows users to add annotations to web pages, but it is often fraught with error and difficult to apply with a huge numbers of web documents. Semi automatic [3] is based on information extraction that is trained to handle structurally and/or linguistically similar documents and allows the user to add or delete annotation. Automatic annotation relies on automatic annotating algorithm.

Many semantic annotation tools have been appeared in this field such as Magpie [19], Shoe [20], Gate and Ontomat. They can annotate paragraph, sentence or term. Magpie and SHOE have a predefined ontology used for the annotation, they cannot import any ontology in a specific domain but Ontomat and gate can import any ontology files.

2. Ontology learning life cycle

The goal of semantic annotation tool is executed through the use of ontology learning framework. Ontologies formalize the in tensional aspects of a domain, whereas the extensional part is provided by a knowledge base that contains assertions about instances of concepts and relations as defined by the ontology. The process of defining and instantiating a knowledge base is referred to as knowledge markup or ontology population, whereas semi-automatic [4] support in ontology development is usually referred to as ontology learning.

Ontology learning, in the Semantic Web context, is primarily concerned with knowledge acquisition from and for Web content and is thus moving away from small and homogeneous data collections to tackle the massive data heterogeneity of the World Wide Web[4].

The following figure [Figure (2)] shows the ontology learning framework proceeds through: ontology import, extraction, pruning, refinement phases.

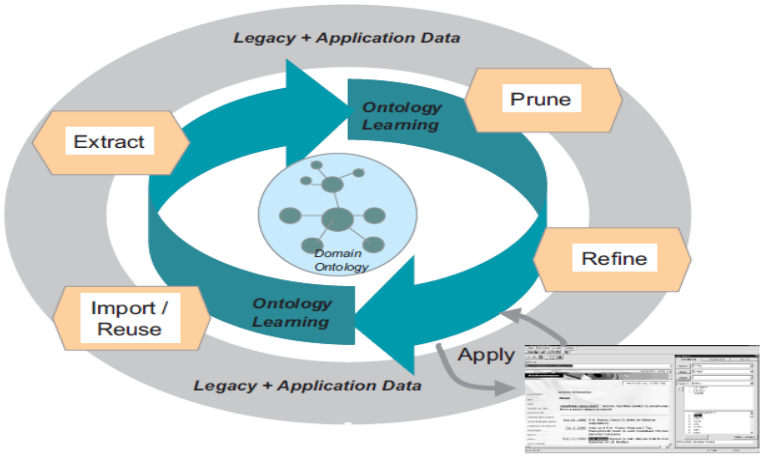

Figure (2): Ontology learning Framework "process"

2.1 Import/Reuse

In order to save time and other resources, it is thus desirable to import and reuse ontologies. This task involves the selection of relevant ontologies. The definition of appropriate imports strategies and the subsequent selection and merging of relevant conceptual structures.

2.2 Extract

In the ontology extraction phase of the ontology learning process, major parts, i.e. the complete ontology or large chunks reflecting a new sub domain of the ontology, are modeled with learning support exploiting various types of (Web) sources. This phase is composed of subtasks which called "ontology learning layer cake".

Ontology learning cake is composed of:

Term Extraction. It implies more or less advanced levels of linguistic processing, i.e. phrase analysis to identify complex noun phrases that may express terms and dependency structure analysis to identify their internal semantic structure [6]. It typically involves methods from the fields of Information Extraction and Information Retrieval. The term Information Extraction [7] refers to a set of techniques and methods used to detect and process information in larger documents and subsequently present it in a structured format.

Synonym Level. It addresses the acquisition of semantic term variants in and between languages, where the latter in fact concerns the acquisition of term translations. Much of the work in this area has focused on the integration of WordNet7 for the acquisition of English synonyms, and EuroWordNet8 for bilingual and multilingual synonyms and term translations [4].

Concept Extraction from Text. The concept induction or formation should provide:

1- An intentional definition of the concept.
2- A set of concept instances, i.e. its extension.
3- A set of linguistic realizations, i.e. (multilingual) terms for this concept.

Concept Extraction Hierarchy. It implies the use of lexicon syntactic patterns which aims to access hyponym lexical relations from text. It uses a set of predefined lexico-syntactic patterns which occur frequently and indicate the relation of interest that can be recognized with little or no pre-encoded knowledge. Related to this are also approaches that exploit the internal structure of noun phrases to derive taxonomic relations between classes expressed by the head of the noun phrase and its subclasses that can be derived from a combination of the head and its modifiers [6].

Relation Extraction. Most of the work on text mining combines statistical analysis with more or less complex levels of linguistic analysis, e.g. by exploiting syntactic structure and dependencies for relation extraction as reported for instance by [6,9,10].

Association Rule Learning Algorithms. They are typically used for prototypical applications of data mining, like finding associations that occur between items, e.g. supermarket products, in a set of transactions, e.g. customers' purchases. The generalized association rule learning algorithm extends its baseline by aiming at descriptions at the appropriate level of the taxonomy, e.g. "snacks are purchased together with drinks" rather than "chips are purchased with beer" and "peanuts are purchased with soda".

2.3 Prune

There are at least two dimensions to look at the problem of pruning. First one needs to clarify how the pruning of particular parts of the ontology (e.g., the removal of a concept or a relation) affects the rest. For instance, Peterson et. al. [8] has described strategies that leave the user with a coherent ontology (i.e. no dangling or broken links). Second one may consider strategies for proposing ontology items that should be either kept or pruned.

2.4 Refine

Similar to the extraction phase refining also tries to add or modify conceptual structures [7] but with the objective to fine tune the target ontology.

The refinement phase may use data that comes from the concrete Semantic Web application, e.g. log files of user queries or generic user data. Adapting and refining the ontology with respect to user requirements plays a major role for the acceptance of the application and its further development.

The same algorithms may be used for extraction as for refinement. However, during refinement one must consider in detail the existing ontology and the existing connections into the ontology.

3. Ontology Learning Techniques

Ontology learning techniques are derived from information extraction, machine learning and natural language processing [7]. They are adapted to be usable by ontology learning applications.

3.1 Natural Language Processing

It deals with the analysis of term "text" in order to make it understandable for machines. To make this analysis it needs a parser that follows these steps:

1- The first step involves tokenization and normalization of the text document. A parser detects sentences and word boundaries and outputs a stream of tokens. The normalization is used to detect instance missing whitespaces between words or ambiguous punctuation. Tokenization and normalization is applied by the use of POS tagging (part of speech tagging) which assigns each token, its respective word category such as noun, adjective, verb and preposition.
2- The second step involves lemmatization which means the reduction of tokens to their base form. Ex: eats, eat, eating their base is "eat". The parser ca also detects the noun phrases "with" ex:"glycans react with human body cells".

3.2 Machine Learning

Approaches from the field of Machine Learning used to solve the problem of classifying a set of documents into categories. They are two kinds of algorithm used supervised and unsupervised ones. Supervised algorithms usually need sample of training data, which is used to construct initial models which will be used to make predictions about test data. Unsupervised algorithms do not incorporate test data but classify the data by grouping similar elements together.

The following figure [Figure (3)] shows on the left hand side applies typical supervised classification where the three classes are known a priori. The training data set is represented by those documents that are already assigned to a class. The problem now is to assign the other documents to one of these classes. On the right hand side applies a typical unsupervised setting. The target of this approach deals with an unknown number of groupings from the data based on a given similarity measure. Documents that are similar to each other should be grouped together.

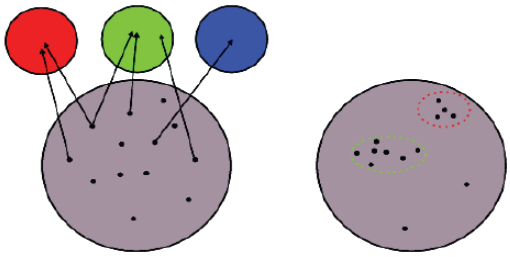

Figure3: Comparison of supervised and unsupervised machine learning approaches. On the left hand side we have the supervised problem setting, on the right hand side the unsupervised one. Figure is taken from Andrews and Fox [2007].

The crucial issue herein is often the definition of a function that calculates this similarity. The unsupervised approaches typically use a clustering algorithm. Clustering is generally seen as the task of finding groups (clusters) of similar objects in data. Available approaches can be classified in hierarchical and non hierarchical ones. Hierarchical algorithms cluster data and additionally order them in a hierarchical structure whereas non hierarchical ones output an unrelated set of flat clusters.

3.3 Vector Space Model

The Vector Space Model (VSM) is widely used in the areas of information extraction and machine learning. It was initially designed as a model to represent arbitrary text documents as vectors from a common vector space [Manning et al., 2008]. However, it can be adapted to work with any type of data. Vector Space Model [11] incorporates local and global information.

Let's assume we have two documents d1 and d2 consisting of a sequence of words. The Goal is now to transform both documents into vectors of the same vector space. As dimensions we use the set of all unique words from both documents. The two documents are:

d1 = the weather is sunny and cloudy.

d2 = the fox is brown and clever.

Table (1) shows all unique words are taken as dimensions of the vector-space and counting the occurrence of each word in the two documents.

Table 1 Dimension of vector space

	The	weather	is	sunny	and	cloudy	fox	brown	clever
d1	1	1	1	1	1	1	0	0	0
d2	1	0	1	0	1	0	1	1	1

The token "the" appears in the two documents while the token "clever" appears only in d2. A common algorithm of the vector-space model is then to calculate the similarity between two or more objects [11].

4. Semantic Annotation Evaluation Tools

4.1 Ontomat

Ontomat-Annotizer (Ontomat for short) is a component-based, ontology-driven annotation tool for Web documents. It is the implementation of CREAM (CREAting Metadata for the Semantic Web), a framework for an annotation environment. In the CREAM method, ontology is represented by statements expressing definitions of DAML+OIL classes and properties. Based on those definitions, annotations are a set of instantiations of classes, attributes and relationships that attached to an HTML document.

Annotations are described through the metadata which are derived from ontology definitions. Such metadata is stated as relational metadata because it contains relationship instances [15, 49].

Ontomat includes an ontology browser for the exploration of the ontology, instances and an html browser that displays the annotated parts of the text. It supports both manual and semi-automatic annotation. The semi-automatic annotation approach is developed in S-CREAM (Semi-automatic CREAM) The semiautomatic metadata creation takes advantage of information extraction techniques to propose annotations to metadata creators. A learnable information extraction component– Amilcare [16] [17] is integrated in S-CREAM.

Ontomat annotizer enables the knowledge editor to enrich their WebPages with DAML metadata instead of manually annotating the page with a text editor, it allows highlighting relevant parts of the webpage and creates new instances via drag drop interaction.

4.2 GATE

GATE is an architecture for language engineering developed at the University of Sheffield [12], containing a suite of tools for language processing, and in particular, a vanilla IE system ANNIE. In traditional IE applications, GATE is run over a corpus of texts to produce a set of annotated texts.

GATE is a leading NLP and IE platform developed in the University of Sheffield, consists of different modules:

1- Tokenizer.
2- Gazetteer.
3- Sentence Splitter.
4- Part-of-Speech Tagger (POS-Tagger).
5- Named Entity Recognizer (NE-Recognizer).
6- OrthoMatcher (Orthographic Matcher).
7- Co reference Resolution.

GATE's IE system is rule-based, which means that unlike machine-learning based approaches, it requires no training data [13]. On the other hand, it requires a developer to create rules manually, so it is not totally dynamic "automatic".

GATE comprises three principal elements [14]

1- A database for storing information about texts and a database schema based on object oriented model of information about texts (the GATE Document manager –GDM).
2- A graphical interface for launching processing tools on data and viewing and evaluating the results (GGI).
3- A collection of wrappers for algorithmic and data resources that interoperates with the database.

GATE architecture distinguishes two basic kinds of resources: Languages resources and processing resource. A language resource can be individual text loaded as GATE document or a collection of texts loaded as GATE corpus. A processing resource is a distinct processing component such as a tokenizer or named entity recognition [13] loaded by the use of an information extraction system called ANNIE.

5. Proposal Framework for Comparative Study to Ontomat and Gate

Semantic annotation tools comparison is based on the use of two ontology files in Biochemistry domain and two web documents in the same domain.

5.1 Input "The ontology files and web documents":

1- GlycO[25]: Includes a classification about Glycans, its reactions, chemical entities. The file format is '.owl'. Glycans are complex carbohydrate structures, which play key roles in the development and maintenance of living cells. Glycans[21] are built from simpler monosaccharide residues (such as mannose and glucose), which

constitute the nodes of tree structures with edges that are comprised of chemical bonds between the residues. The synthesis of these glycans in organisms is an intricate process that can be modeled as a collection of biosynthetic pathways. At each step in such a pathway, an enzyme-catalyzed reaction 'adds' a new residue as a leaf to an existing structure or 'moves' a whole subtree to a different parent. the web document[23] is used to enrich the GlycO ontology with additional information.

2- EnzyO[26] : Enzyme activity plays a crucial role in the synthesis of glycans,they are subset of proteins. Enzyme ontology[22] EnzyO keeps track of enzymes that catalyze the actions which produces the glycan structures. The ontology keeps track of basic information about enzymes for example their enzyme commission number (EC) , their protein structure as well as associations with genes that codes for it and the reactions it participates in. the web document[24] is used to enrich the EnzyO ontology with additional information.

5.2 Points of comparison that are used to point out between Ontomat and GATE are:

1- Input.
2- IE information extraction.
3- Ontology learning techniques.
4- Annotation type.
5- Ontology refinement.
6- Output.

Table (2) shows the comparison between input, information extraction and ontology learning techniques. For the input: GlycO and EnzyO are owl files which include xml tags. To be usable by GATE they must be converted to ".rdf-xml " format ,the conversion is applied by the use of online owl syntax converter[27] or Protégé program. The information extraction in Ontomat is applied with ".wsdl" web ontology file. It can be enhanced by the application of Amilcare toolkit with html web pages. GATE applies two techniques of ontology learning, it applies the natural language processing by the use of Annie English tokenizer, POS tagging, NE transducer and Wordnet.

.

Table 2 Tools comparison with input, information extraction and ontology learning techniques

Input	Tools	
	Ontomat	**GATE**
Ontology file	Accepts the ontology file with the following format : ".owl" ".n3"	Accepts the ontology file with the following format : • rdf-xml • ntriples • n3 • .turtle
Web Page URL	Accepts the web page with ".html" Also accepts the ".wsdl" format which is a web ontology file.	Accepts the web page with ".html"
Information Extraction	Applies the information extraction algorithm by the use of "Amilcare toolkit" which is used only with ".wsdl" web ontology file [18] as input and is not available to use it with html web page.	Uses Annie information extraction algorithm which applies the following modules: • Annie English tokenizer. • Pos tagger • Gazetteer. • Named entity recognition
Ontology Learning techniques	Uses the machine learning technique "wrapper induction".	Uses the machine learning technique"wrapper" and Natural language processing.

Table 3 Tools comparison with Annotation type

Annotation Type	Tools	
	Ontomat	**GATE**
Manual	√	X
Semi automatic	√	√
Automatic	X	X

Table (3) shows that both of Ontomat and GATE are semi automatic tools that allow to the knowledge editor to add annotations via a graphical user interface. If the ontology includes a large amount of classes and subclasses, it will be more difficult to annotate a word in the web document with an ontology class. The generation of automatic suggestions will facilitate the annotation.

Table (4) shows the ontology refinement and the output for each tool. For the ontology refinement, GATE can be enhanced by the use of an algorithm that allows to the knowledge editor to create a relationship between main class and subclasses. Concerning the output, the imported webpage in GATE can be saved as html but without annotation.

Table 4 Tools comparison with ontology refinement and output

	Tools	
	Ontomat	**GATE**
Ontology Refinement	The knowledge editor can add/edit new instances, attributes, new superclasses and subclasses. Also he can add/edit a relationship between subclasses and also can add a relationship between the main class and other classes.	The knowledge editor can add/edit new instances, attributes, new superclasses and subclasses. Also he can add/edit a relationship "object property" between subclasses. But the interface prevents the knowledge editor to create a relationship between the main class and other subclasses although the main class is predefined in the ontology file but the interface can't detect it.
Output (ontology file/ web page)	An updated ontology file which is enriched by the new annotation. Web page saved as html file, its source includes the html tags and the new annotation which is compatible with web agents.	An updated ontology file which is enriched by the new annotation Web page saved as xml file, its source includes the xml tags and annotation tags.

6. Conclusions

Both of Ontomat and gate follow the ontology learning life cycle. GATE applies the use of natural language technique while Ontomat doesn't apply it. In future work, refinement phase can be enhanced by the use of an algorithm which detects tokens in the imported web page document which is found in the ontology classes , the use of association rules that enrich the annotation of the webpage".html" can enhance the ontology learning cycle.

References

[1] Z.ahmed DIPLOMA THESIS domain specific Information Extraction for Semantic Annotation,2009.

[2] T. Gruber. Towards principles for the design of ontologies used for knowledge sharing. Int. J. of Human and Computer Studies, 43:907–928, 1994

[3] S. handschuh,S.taab, and R.studer, Leveraging Metadata Creation for the Semantic Web with CREAM,2003.

[4] P. Buitelaar, P.Cimiano and B. Magnini ,Ontology Learning from Text: An Overview,2005.

[5] A. Maedche and S.Staab, Ontology Learning for the Semantic Web Institute AIFB, D-76128 Karlsruhe, Germany,2001.

[6] P. buitelaar, P.Cimiano and B. Magnini, Ontology learning from text: methods evaluation and application,2003.

[7] C. fischer Ontology Learning for Enabling Product Comparisons on theWeb,2010.

[8] M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In Proceedings of COLING-92, Nantes, France, 1992.

[9] J M. Ciramita, A. Gangemi, E. Ratsch, J. Saric, and I. Rojas. Unsupervised learning of semantic relations between concepts of a molecular biology ontology, 2005.

[10] P. Gamallo, M. Gonzalez, A. Agustini, G. Lopes, and V. S. de Lima. Mapping syntactic dependencies onto semantic relations,2002.

[11] Salton, Gerard. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

[12] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications, 2003.

[13] D. Maynard, V. Tablan, and H. Cunningham. NE recognition without training data on a language you don't speak. In ACL Workshop on Multilingual and Mixed-language Named Entity Recognition: Combining Statistical and Symbolic Models, Sapporo, Japan, 2003.

[14] H.Cunningham, Y.Wilks, R.Gaizauskas GATE - General Architecture for Text Engineering, 2002.

[15] A.Limi,A.Runyan andV.andersen.Ontomat tutorial tool review,2004.

[16] F.Ciravegna, A.Dingli, Y.Wilks, and D.Petrelli.Adaptiveinformation extraction for document annotation in amilcare. In Proc. of the 25thannual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2002), pages 451–451, New York, NY, USA, 2002.ACM Press.

[17] S.Handschuh, S.Staab, and R. Studer. Leveraging metadata creation for the semantic web with CREAM. In Proc. of the Annual German Conference on Advances in Artificial Intelligence (KI 2003), pages 19–33, Berlin. LNCS2821, Springer, 2003.

[18] S.Agarwa, S.Handschuh, and S.Staab, Surfing the Service Web, International Semantic Web Conference - ISWC, 2003.

[19] J.Domingue,M.Dzbor and E.Motta. Magpie: Browsing and navigating on the semantic web. Proceedings ACM Conference on Intelligent User Interfaces (IUI), pages 191-197, January 2004.

[20] S. Luke, L. Spector, D. Rager, and J. Hendler. Ontology-based Web agents. In Proceedings of the First International Conference on Autonomous Agents, pages 59–66. ACM, 1997.

[21] OBO: Open Biomedical Ontologies. http://obo.sourceforge.net.

[22] I.Măndoiu,R.Sunderraman,A.Zelikovsky.Bioinformatics research and applications. In proceedings of the fourth international Symposium,page.309-309, Atlantaa,GA, USA,ISBRA 2008.

[23] http://www.glycominds.com/Science.asp?medical=1184

[24] http://www.cazy.org/GT94.html

[25] http://lsdis.cs.uga.edu/projects/glycomics/2006/GlycO.owl

[26] http://lsdis.cs.uga.edu/projects/glycomics/2006/EnzyO.owl

[27] http://owl.cs.manchester.ac.uk/converter/