Chapter VI

A Comparison of Semantic Annotation Systems for Text-Based Web Documents

Lawrence Reeve, Drexel University, USA

Abstract

The Semantic Web promises new as well as extended applications, such as concept searching, custom Web page generation, and question-answering systems. Semantic annotation is a key component for the realization of the Semantic Web. The volume of existing and new documents on the Web makes manual annotation problematic. Semi-automatic semantic annotation systems, which we call platforms because of their extensibility and composability of services, have been designed to alleviate this burden for text-based Web documents. These semantic annotation platforms provide services supporting annotation, including ontology and knowledge base access and storage, information extraction, programming interfaces, and end-user interfaces. This chapter defines a framework for examining semantic annotation platform differences based on platform characteristics, *surveys several recent platform implementations, defines a classification scheme based on information extraction method used, and discusses general platform architecture.*

Introduction

The Semantic Web, in Berners-Lee (1998), is the next generation of the Web providing machine-understandable information that is based on meaning. One way to provide meaning to Web information is by creating ontologies, and then linking information on a Web page to specifications contained in the ontology using a markup language (Berners-Lee et al., 2001). A key problem for the realization of the Semantic Web is providing these annotations for both existing and new documents on the Web. Semantic annotation is the process of mapping instance data to an ontology. Ontologies are conceptualizations of a domain that typically are represented using domain vocabulary (Chandrasekaran, Josephson, & Benjamins, 1999). Benefits of adding meaning to the Web include: query processing using concept-searching rather than keyword-searching (Berners-Lee et al., 2001); custom Web page generation for the visually-impaired (Yesilada, Harper, Goble, & Stevens, 2004); using information in different contexts, depending on the needs and viewpoint of the user (Dill et al., 2003); and question-answering (Kogut & Holmes, 2001).

It is not yet possible to automatically identify and classify all entities within source documents with complete accuracy (Popov et al., 2003). Manual annotation can be done using tools such as Semantic Word (Tallis, 2003), which provides a single interface for authoring as well as document markup. Manual approaches, however, suffer from several drawbacks. Human annotators can provide unreliable annotation for many reasons: complex ontology schemas, unfamiliarity with subject material, and motivation, to name a few (Bayerl, Lüngen, Gut, & Paul, 2003). It is expensive to have human annotators markup documents (Cimiano, Handschuh, & Staab, 2004), and the human annotator may not consider using multiple ontologies (Dingli, Ciravegna, & Wilks, 2003). Documents and ontologies can change, requiring new or modified markup, which leads to document markup maintenance issues (Dingli et al., 2003). Finally, the volume of existing documents on the Web can lead to an overwhelming task for humans to manually complete (Kosala & Blockeel, 2000). For all these reasons, manual efforts have been identified as a "knowledge acquisition bottleneck" (Maedche & Staab, 2001). Semi-automatic annotation platforms offer advantages over manual efforts, primarily document volume scalability through reduction of the human workload (Dill et al., 2003).

Semi-automatic semantic annotation systems, which we call Semantic Annotation Platforms (SAPs), have been developed to overcome the scalability issue of providing annotations for the large number of documents on the Web. Semi-automatic annotation of text documents can be seen as a typical information extraction for named-entity recognition process, but is different in that type information from a rich ontology is more specific and also the entities must be clearly identified and not just recognized as an entity of some type, as is the case with basic information extraction efforts (Popov et al., 2003). Semi-automatic annotation has been developed using research done in the areas of information extraction, information integration, wrapper induction, and machine learning (Dingli et al., 2003).

The remainder of this chapter defines a framework for viewing the differences between platforms, provides an overview of a few representative platforms, and briefly analyzes each platform using the platform characterization framework. This chapter expands on our initial work (Reeve & Han, 2005), which classified and surveyed several platforms, but did not fully identify the platform characteristics framework and platform classification scheme that the survey was based on, or identify a general semantic annotation platform architecture. The platform characteristics framework and classification system is useful for evaluating potential applications of a semantic annotation platform, where each has its own strengths based on domain, document structure, information extraction methods, ontology support, and manual effort required. The platform architecture is useful for understanding how these multiservice platforms are composed.

Development of Semantic Annotation Platforms

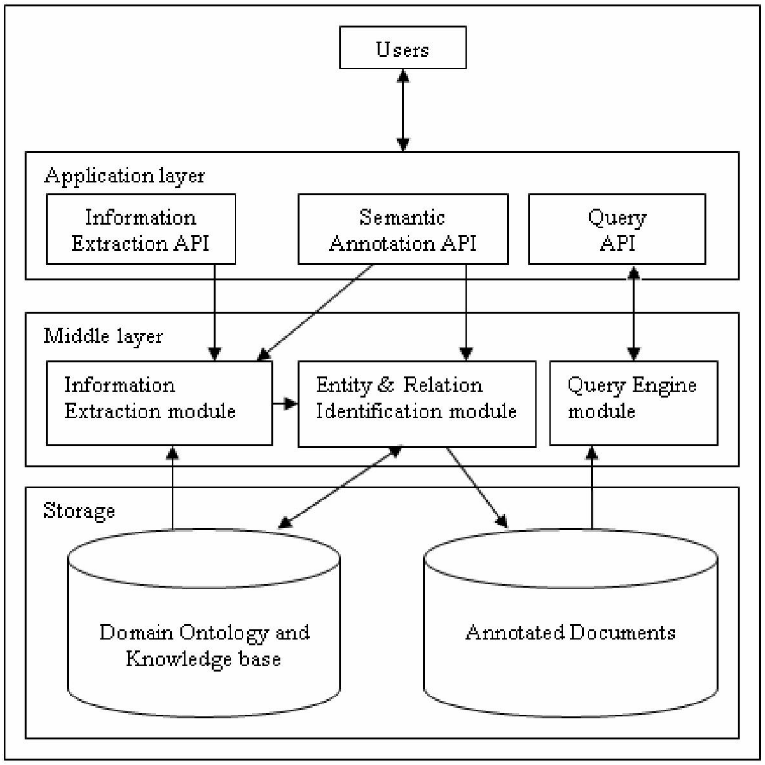
The semi-automatic annotation systems compared in this chapter provide semantic annotation of text-based Web documents. We call such systems platforms, largely due to their extensibility and composability; some of the literature also refers to them as platforms (Popov et al., 2003; Dill et al., 2003). There have been many efforts to build platforms for semantic annotation. Several of these platforms are briefly discussed in the *Platform Overviews* section, and were selected as a representative sample, rather than an exhaustive list, of the platform classifications discussed in the *Platform Classification* section. These semantic annotation platforms are diverse enough to show the distinguishing platform characteristics identified in the *Platform Characteristics* section.

Platform Architecture

Figure 1 shows the general architecture of a semantic annotation platform (SAP) as a composable system. Most SAPs are extensible, meaning various components can be replaced with alternate implementations. The advantage of an extensible annotation platform is that it can be adapted to serve many needs, such as changing domains, languages, or providing scalability.

The Application layer is responsible for providing an end-user interface to the annotation services provided by a SAP. Examples include facilities for annotating a document or document set and then potentially confirming the annotations before committing them, providing a query interface for searching annotations, and providing a user interface for configuring the information extraction component. The Application layer is the primary application programming interfaces (API) layer. A set of general programmatic interfaces designed to shield the applications from changes in the middle layer are defined in this layer. Applica-

Figure 1. General architecture of a semantic annotation platform


Users
Application layer
Information Extraction API
Semantic Annotation API
Query API
Middle layer
Information Extraction module
Entity & Relation Identification module
Query Engine module
Storage
Domain Ontology and Knowledge base
Annotated Documents

tions call the defined APIs in order to perform actions on behalf of an application and they can be quite numerous, covering annotation, information extraction, search, storage management, and many other provided services. The middle layer contains the actual components that perform work for an application, such as information extraction for concept (names and relationships) identification. The Application layer provides a consistent view to an application, but the middle layer is tied to an existing or adapted tool. For example, the information extraction component may switch from a pattern-based tool to a statistical tool, and it is unlikely the programmatic interface is the same for both. Finally, the Storage Layer is designed to provide storage and storage management facilities for storing long-term data such as ontologies, document annotations and knowledge bases.

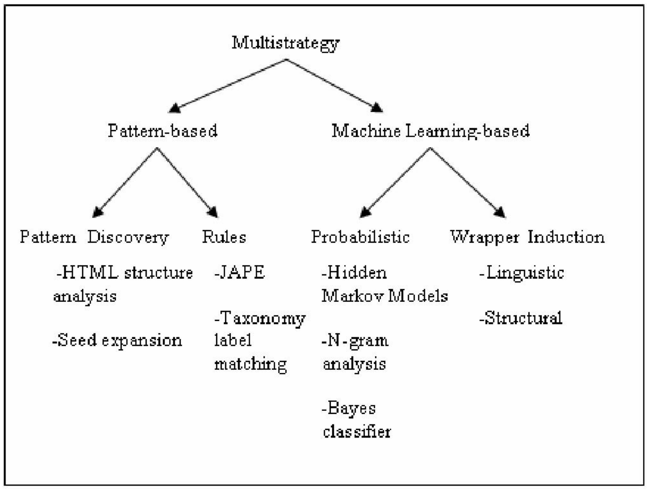
Platform Classification

Current semantic annotation platforms use several methods for information extraction (IE) from Web documents. Figure 2 shows a hierarchical classification of annotation platforms based on their IE component. This classification scheme can be used to organize the platforms performing semantic annotation. While semantic annotation platforms have many aspects, the information extraction approach currently used to find entities within text has the most impact on the effectiveness of the platform. For this reason, the information extraction (IE) approach of each platform is used to organize the platforms.

The top-level approach is multistrategy, which uses a combination of the lower level approaches. A platform using a multistrategy approach is able to adapt its IE methods based on the text it is processing in order to obtain the best results. The multistrategy approach uses a high-level identification of text genre, and then executes the appropriate IE methods (either pattern-based or machine learning-based). No semantic annotation platform to date is using a complete multistrategy approach incorporating both pattern and machine learning approaches.

The two components levels composing the multistrategy approach are pattern-based methods and machine-learning methods. Pattern-based methods are systems composed of manual rules. The rules are typically hand-crafted rules or seed patterns that define how entities can be found in text. A limiting factor on the scalability of such systems is that the manual rule generation process can be maintenance-intensive. Each time a data source changes, the pre-defined rules may also need to be changed. Machine-learning approaches are mainly based on supervised learning and use pre-annotated examples to learn how to identify

Figure 2. Classification of semantic annotation platforms based on information extraction method used



entities. Rules are learned from a pre-annotated corpus, and then later applied to perform annotation. The implication is that sufficient training data exists to adequately train the system. Hidden Markov Model (Rabiner, 1989) is an example of a machine-learning approach that can be used. The Hidden Markov Model approach is not currently being used by any of the semantic annotation platforms as an information extraction method, but has been used successfully to extract attributes from text (Svab, Labsky, & Svatek, 2004).

Pattern-Based Methods

Patterns are widely used in semantic annotation platforms. **Pattern discovery** works by taking a few seed samples, finding entities based on the patterns, expanding the seed samples with patterns from the new entities found, and repeating the process until no more instances are found, or the user stops the iterative process (Brin, 1998). Patterns can exploit known linguistic patterns, such as Hearst patterns (Hearst, 1992), to find entities in text.

Rules can be manually-generated or they can be learned using machine-learning techniques. For classification purposes, manual rule generation is considered as part of the pattern-based classification, as shown in Figure 2. The reason is because the rules are initially manually specified by the user. Rules can take many forms. They can be as simple as labels, as in a gazetteer, or they can be complex definitions expressed in a grammar using a tool such as the Java Annotations Pattern Engine (Maynard, 2003). In general, rule-based systems do not perform as well as machine-learning based systems. However, adaptive rule platforms, such as the MUSE platform (see the *Platform Overviews* section), perform competitively with machine learning-based platforms.

Machine Learning-Based Methods

Platforms based on machine learning are divided using two approaches: probabilistic and wrapper induction. The more common of the two approaches is wrapper induction, most likely because wrapper induction excels at extraction of text from structured documents, and is a more quickly solvable problem than extraction of entities from unstructured text.

Probabilistic methods use algorithms such as Hidden Markov Models (Rabiner, 1989) to perform information extraction. Probabilistic approaches are not yet widespread in semantic annotation work.

Wrapper induction is a way to automatically construct a procedure for extracting content from a particular resource, such as a Web page. A wrapper is a function from a page to the set of tuples it contains (Kushmerick, Weld, & Doorenbos, 1997). Wrappers are used when there exists a repeatable structure to extract information from. Many web sites have pages generated using from a back-end database, and the pages generated follow a common template. Wrappers reverse the page generation process to retrieve the original database tuples. Wrappers can be hand-crafted, or they can be learned. Manual wrappers require the user to mark areas of interest within a document. The machine can then extract entities from documents with a similar structural format as the manually marked-up document (Vargas-Vera et al., 2002). Kushmerick (1997) defined a method for performing wrapper induction, where the wrappers are automatically learned from example query responses from a data source. Wrappers are most effective when the data is presented in a structured format, such as product catalogs (Dingli et al., 2003).

Wrappers can also be linguistic-based, where the wrapper induction process discovers linguistic rules for identifying entities (Vargas-Vera et al., 2002). Amilcare (University of Sheffield, 2002) implements the LP^2 algorithm (Ciravegna, 2001), which performs rule induction using both linguistic and structural information.

Platform Overviews

This section provides an overview of each of the semantic annotation platforms surveyed. For further details of each system, readers are recommended to refer to our extended technical report (http://www.pages.drexel.edu/~lhr24/) and previous paper (Reeve & Han, 2005). The performance as reported by each platform author is then discussed and compared.

AeroDAML

AeroDAML (Kogut & Holmes, 2001) is designed to map proper nouns and common relationships to corresponding classes and properties in the DARPA Agent Markup Language (DAML) (DARPA, 2004) ontologies. AeroDAML uses AeroText (Lockheed Martin, 2005) for its information extraction component. The default ontology used by AeroDAML consists of two parts. The upper level ontology uses the WordNet noun synset hierarchy, while the lower level ontology uses the knowledge base provided by AeroText (Kogut & Holmes, 2001).

AeroText consists of four main components: (1) a Knowledge Base (KB) compiler for transforming linguistic data into a run-time knowledge base; (2) a Knowledge Base Engine for applying the KB to source documents; (3) a development environment for building and testing KBs, and (4) a Common Knowledge Base containing domain independent rules for extracting proper nouns and relations.

Armadillo

Armadillo (Dingli et al., 2003) is used to mine home pages of computer science faculty to find personal contact information, such as name, position, home page, and e-mail address. Armadillo uses a pattern-based approach to find entities, but is unique in that it finds its own initial set of seed-patterns, rather than requiring an initial set of seeds, as described in Brin (1998). Once the seeds are found in the corpus, pattern expansion is then used to discover additional entities. The seed discovery and expansion finds faculty names in Web pages. Since many names may be discovered, Web services, such as Google and CiteSeer, are queried to provide evidence a person actually works in the computer science department. The names are then used to discover home pages, where detailed information about a person can often be found and extracted. The idea of information redundancy is exploited to verify entities that have been extracted.

KIM

The Knowledge and Information Management (KIM) platform (Popov et al., 2003) contains an ontology, knowledge base, a semantic annotation, indexing and retrieval server, as well as front-ends for interfacing with the server. For ontology and knowledge base storage it uses the SESAME RDF repository (openRDF.org, 2005), and for search it uses a modified version of the Lucene (Cutting, 2004) search engine. The semantic annotation process relies on a prebuilt lightweight ontology called KIMO as well as an interdomain knowledge base. KIMO defines a base set of entity classes, relationships, and attribute restrictions. The knowledge base is populated with 80,000 entities consisting of locations and organizations, gathered from a general news corpus. Named-entities found during the annotation process are matched to their type in the ontology and also to a reference in the knowledge base. The dual mapping allows the information extraction process to be improved by providing disambiguation clues based on attributes and relations.

The information extraction component of semantic annotation is performed using components of the GATE (H. Cunningham, Maynard, Bontcheva, & Tablan, 2002) toolkit. GATE provides IE implementations of tokenizers, part-of-speech taggers, gazetters, pattern-matching grammars (JAPE), and coreference resolution (Popov et al., 2003). Some components of GATE have been modified to support the KIM server. The gazetteer, for example, performs entity alias lookups using the knowledge base rather than an external file.

MnM

MnM (Vargas-Vera et al., 2002) provides an environment to manually annotate a training corpus, and then feed the corpus into a wrapper induction system based on Amilcare (University of Sheffield, 2002). Once the platform has been trained and rules have been induced from a training corpus, the system annotates text documents based on user-defined semantic tags. Each document is annotated, presented to the user for approval, and sent to the ontology server to populate the ontology with instance data. Populating the ontology with instance data is done by taking the data from the semantic annotations and populating each ontology entry with its attribute values. The population phase will attempt to provide values for as many attributes as possible. If the semantic annotation process does not provide all attribute values, the user will need to manually provide attribute values. This can occur, for example, if the attribute value is not mentioned in the text, or if the annotation rule set is incomplete and needs more training.

MUSE

MUSE (Maynard, 2003) uses an adaptive rule-based approach to perform annotation. Text attributes are used to conditionally run various processing resources, such as different gazetteers, over a document. Semantic tagging is accomplished using the Java Annotations Pattern Engine, also known as JAPE (Cunningham, Maynard, & Tablan, 2000). MUSE has been used to perform annotation in languages other than English to demonstrate its adaptability and compare its performance with machine learning systems, which typically require large training data. Since MUSE is modular, only the language-dependent parts needed to be converted between languages. For example, the part-of-speech tagger, gazetteer lists, and potentially language-dependent parts of the semantic tagger may need to be modified. The multiple language adaptation projects demonstrated that MUSE provides an advantage over machine learning based approaches because it requires a smaller amount of training data.

Ont-O-MAT using Amilcare

Ont-O-Mat is an implementation of the S-CREAM (Semi-automatic CREAtion of Metadata) semantic annotation framework (Handschuh, Staab, & Ciravegna, 2002). The information extraction component is based on Amilcare. Amilcare is machine-learning based and requires a training corpus of manually annotated documents. Amilcare uses the ANNIE (A Nearly-New IE system) part of the GATE toolkit to perform information extraction tasks such as tokenization, part-of-speech tagging, gazetteer lookup, and named-entity recognition (Handschuh et al., 2002). The result of ANNIE processing is passed to Amilcare, which then induces rules for information extraction using a variant of the LP^2 algorithm. The wrapper induction process uses linguistic information, and is the same Amilcare wrapper induction process as MnM (Vargas-Vera et al., 2002) uses, generating tagging and correction rules.

Ont-O-MAT using PANKOW

Ont-O-Mat provides an extensible architecture that allows replacement of selected components. The original Ont-O-Mat implementation was done using Amilcare. In this case, Ont-O-Mat replaces the annotation portion with a implementation of the PANKOW (Pattern-based Annotation through Knowledge On the Web) algorithm (Cimiano et al., 2004). The PANKOW process takes proper nouns from the information extraction phase and generates hypothesis phrases based on linguistic patterns and the specified ontology. For example, a sports ontology may generate hypothesis phrases from the proper noun "Pete Rose" using patterns such as "Pete Rose *is a* Player" and "Pete Rose *is a* Team," where "Player" and "Team" are ontology concepts. The hypothesis phrases are then presented to the Google Web service. The phrase with the highest query result count is then used to annotate the text with the appropriate concept. The core principle is called "disambiguation by maximal evidence" (Cimiano et al., 2004). This principle is similar to the approach used by Armadillo (Dingli et al., 2003), which uses multiple web services to find maximal evidence.

SemTag

SemTag is the semantic annotation component of Seeker, a comprehensive platform for performing large-scale annotation of Web pages (Dill et al., 2003). SemTag has been applied as a non-domain specific semantic annotation tool. It has annotated 264 million Web pages, generating 434 million automatically disambiguated semantic annotations. The taxonomy used by SemTag is TAP. TAP is shallow and covers a range of lexical and taxonomic information about popular items such as music, movies, authors, sports, health, and so forth. The annotations generated by SemTag are stored separate from the source document. It is assumed that the source document is read-only, since the annotator in this case is not the original author. The intent of the SemTag/Seeker design is to provide a public repository with an API that will allow agents to retrieve the web page from its source and then request the annotations separately from a Semantic Label Bureau. SemTag performs annotation by finding terms in the text corresponding to entries in the knowledge base, and then using a novel algorithm called Taxonomy-based Disambiguation to determine a terms position in the taxonomy.

Platform Performance

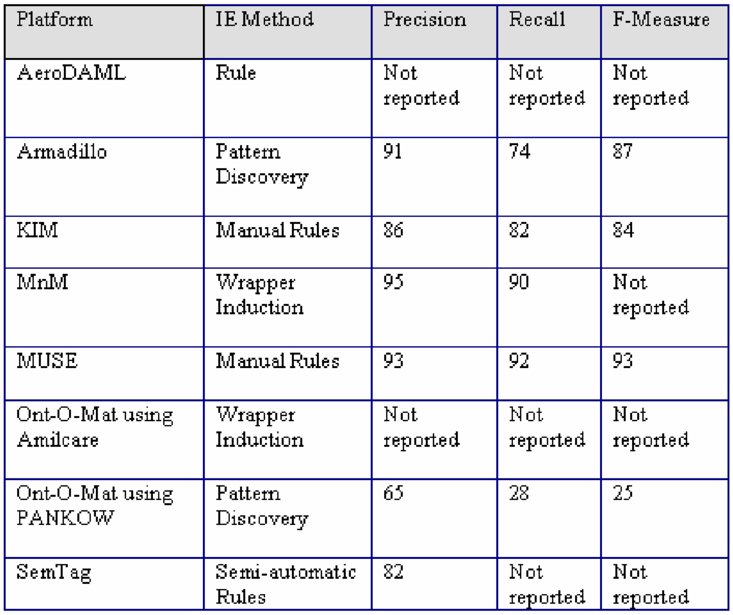
Table 1 shows the author-reported performance of various platforms, with the exception of AeroDAML, Ont-O-Mat using Amilcare, and SemTag, whose authors provided incomplete performance information. The systems were evaluated by the platform authors, using different corpora in sometimes different domains, so direct comparisons are not possible, but the results should give some idea of the performance of each system. The standard measures of Precision and Recall, taken from the information retrieval field, were used by the remaining SAP authors in determining annotation effectiveness. In the general definition of recall and precision shown below, "accurate" and "inaccurate" refer to annotations generated semi-automatically by a SAP, while "all" refers to all annotations generated by a human annotator.

$$Annotation\ \mathrm{Re}call = \frac{accurate}{all}$$

$$Annotation\ \mathrm{Pr}ecision = \frac{accurate}{accurate + inaccurate}$$

The highest performing machine learning-based platform is MnM, which uses Amilcare to take advantage of both structural and linguistic information within a document. For pattern-based platforms, MUSE performs best, most likely because it is adaptive to different text types and its rules have been hand-tuned for the domain it is annotating. The worst performing is Ont-O-Mat using PANKOW, which is a recent effort to use unsupervised learning with linguistic patterns.

Table 1. Information extraction methods and performance measurements of semantic annotation platforms, as reported by the platform authors

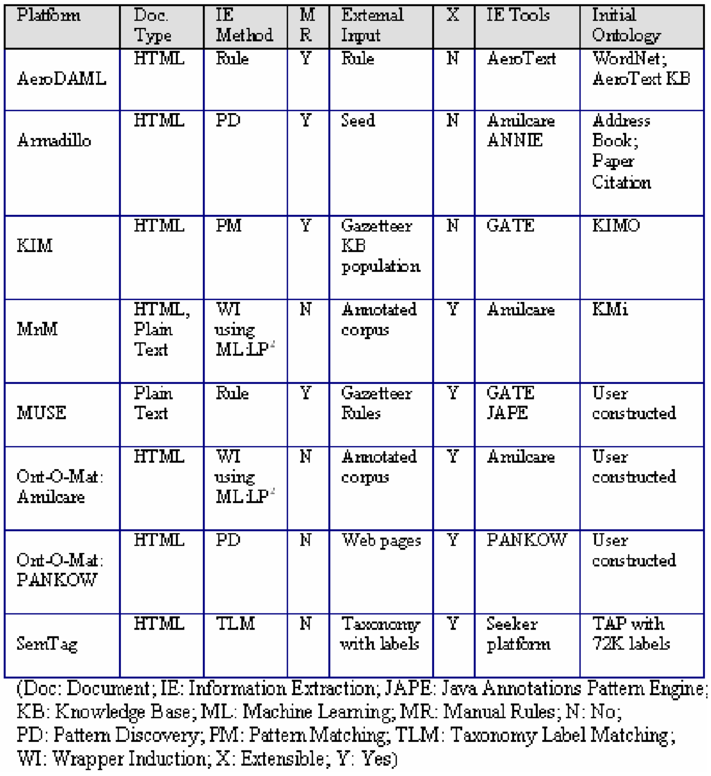


Platform	IE Method	Precision	Recall	F-Measure
AeroDAML	Rule	Not reported	Not reported	Not reported
Armadillo	Pattern Discovery	91	74	87
KIM	Manual Rules	86	82	84
MnM	Wrapper Induction	95	90	Not reported
MUSE	Manual Rules	93	92	93
Ont-O-Mat using Amilcare	Wrapper Induction	Not reported	Not reported	Not reported
Ont-O-Mat using PANKOW	Pattern Discovery	65	28	25
SemTag	Semi-automatic Rules	82	Not reported	Not reported

Platform Characteristics

Figure 1 shows the various layers of abstraction in the design of a Semantic Annotation Platform (SAP). At each layer, the implementer must make decisions which impact the performance and effectiveness of the level based on a set of design goals. These goals are considerations platforms must take into account. As shown in Table 2 and briefly discussed in the *Platform Overviews* section, several SAPs have been developed, and each SAP was designed to address a slightly different annotation need. Table 2 shows some key characteristics for each platform. These characteristics were identified by examining the literature for each platform and noting what characteristics helped distinguish each SAP. We consider the characteristics shown in Table 2 to be a minimum of elements to consider when gaining a sense of a SAP's strengths and weaknesses. In this section, the primary SAP characteristics and their application to the platforms are discussed.

Table 2. Characteristics of semantic annotation platforms



Platform	Doc. Type	IE Method	M R	External Input	X	IE Tools	Initial Ontology
AeroDAML	HTML	Rule	Y	Rule	N	AeroText	WordNet; AeroText KB
Armadillo	HTML	PD	Y	Seed	N	Amilcare ANNIE	Address Book; Paper Citation
KIM	HTML	PM	Y	Gazetteer KB population	N	GATE	KIMO
MnM	HTML, Plain Text	WI using ML-LP^2	N	Annotated corpus	Y	Amilcare	KMi
MUSE	Plain Text	Rule	Y	Gazetteer Rules	Y	GATE JAPE	User constructed
Ont-O-Mat: Amilcare	HTML	WI using ML-LP^2	N	Annotated corpus	Y	Amilcare	User constructed
Ont-O-Mat: PANKOW	HTML	PD	N	Web pages	Y	PANKOW	User constructed
SemTag	HTML	TLM	N	Taxonomy with labels	Y	Seeker platform	TAP with 72K labels

(Doc: Document; IE: Information Extraction; JAPE: Java Annotations Pattern Engine; KB: Knowledge Base; ML: Machine Learning; MR: Manual Rules; N: No; PD: Pattern Discovery; PM: Pattern Matching; TLM: Taxonomy Label Matching; WI: Wrapper Induction; X: Extensible; Y: Yes)

Document Type

The document type shows the type of input typically presented to the platform. Some platforms support many document types, while others only a single or several. Most efforts at Semantic Web annotation are focused primarily on HTML because of its ubiquity, although other document formats such as XHTML, SGML, XML, RTF and others are also supported. HTML presents a challenging environment for entity extraction because HTML markup is designed for presentation and not data manipulation. Unless documents are annotated by customized XML tags with ontological concepts, it is hard to interpret semantics on the documents. While machines can effectively render the format, they cannot easily provide semantic interpretation of it. Applying information extraction methods to the goal of identifying entities and relationships in Web documents requires that information extraction systems be adapted to this environment. Document structure is classified as unstructured, semistructured or structured. Unstructured documents consist of natural language text without any intended structure, such as magazine and journal articles. An information extraction method using natural language processing techniques is needed to parse such documents. Structured and semistructured documents are usually generated documents from back-end content management systems using templates (Mukherjee, Yang, & Ramakrishnan, 2003). Wrapper induction techniques can be used to deduce the location of entities within a document effectively if the structure is well-defined (Kushmerick et al., 1997). All surveyed platforms except MUSE work with HTML documents. MUSE works with plain text documents, as does MnM.

Information Extraction Method

The information extraction method used is a key characteristic in the performance of a SAP, as it is the component which identifies entities in text. Most platforms use a pattern-based approach. Armadillo uses pattern discovery with an initial seed set that is expanded by analyzing HTML structures, such as lists and tables, to find entities. Onto-O-Mat using PANKOW uses linguistic patterns to find maximal evidence to confirm the type of an entity. KIM uses natural language processing to locate entities and then match them against entries in a knowledge base. AeroDAML also uses natural language processing to locate entities within text. SemTag uses a simple approach of tagging entities in the text which match entries in the knowledge base, and in a later step disambiguating the entity tags which may fall into multiple places in the ontology. MUSE uses an adaptive approach, where rules are applied during processing based on attributes found in the text, such as language, domain, and so forth. Both MnM and Onto-

O-Mat use Amilcare to perform entity identification. Amilcare uses a machine learning, rule induction approach. Table 1 shows MnM achieves very good performance among machine learning approaches, while MUSE achieves good performance among pattern-based approaches.

External Input

All systems require some type of manual effort in order to begin the process of semantic annotation. Rules, gazetteers and knowledge bases are common methods of implementing semantic components. Rules can be specified using a grammar such as JAPE (Cunningham et al., 2000), or can be a small set of initial seed patterns to facilitate pattern discovery (Brin, 1998). Gazetteers are lookup lists that map specific literal strings into a semantic concept. The string mappings are predefined by a user for a specific domain. Knowledge bases contain additional entity information than can be stored in the ontology alone. For example, the City of New York entity can have several literal expressions, such as [NYC], [N.Y.C], and [New York] (Kiryakov et al., 2003). Alternatively, knowledge bases can store a large number of additional entity details. For example, in the KIM platform which targets news article annotation, the knowledge base has been prepopulated with about 80,000 entities (Popov et al., 2003).

There exist several problems common to manual external input, such as manual rules, gazetteers, and knowledge bases. First, manual effort is still required to develop the external input to semantic components, although this effort can be spread over many thousands of documents, which is still substantially less than annotating each document manually. Second, the rules and gazetteers must be changed to accommodate different domains and languages (Maynard, 2003). To overcome some of these problems, semantic annotation platforms often incorporate approaches to alleviate this manual bottleneck by automating the construction of the external input. For example, the KIM system incorporates a mechanism where the knowledge base is continually expanded with new instance data, which is verified against a manually annotated corpus and a manually constructed smaller knowledge base (Popov et al., 2003). To alleviate manual construction of rules, several systems use rule induction in order to automatically learn rules to process document text. For example, both the Ont-O-Mat (Handschuh et al., 2002) and MnM (Vargas-Vera et al., 2002) platforms use the LP^2 algorithm from the natural language processing community to learn rules. The algorithm is a wrapper induction algorithm that uses linguistic data as well as structural data, and is based on facilities within the Amilcare toolkit (Vargas-Vera et al., 2002). There has also been work done in the information extraction community using wrapper induction for structural data in order to extract entities from structured or semistructured sources (Kushmerick et al., 1997). A drawback with the wrapper induction approaches, as with any machine learning approach, is that enough training samples must be provided to achieve the desired accuracy. Interestingly, the developers of the MUSE system report that they can move their rule-based system to different languages more quickly than machine-learning based systems (Maynard, 2003). The MUSE system requires that language-specific components, such as gazetteers, be changed to support new languages, but in contrast to machine learning-based approaches, does not require a substantial amount of training data with new languages to achieve high accuracy. The existing semantic rules can be re-used, greatly easing the support of new languages.

Extensibility

Extensible platforms allow for various components to be exchanged. For example, the middle layer of a SAP in figure 1 contains the critical components for semantic annotation performance. In an extensible SAP, a rule induction for information extraction (IE) component can be substituted with a statistical one. The benefits of the rest of the platform components continued to be used while newer IE components are evaluated and integrated. Another advantage is allowing the SAP to adapt to different domains where the information extraction component may perform differently based on the domain document input, as demonstrated by the MUSE system (Maynard, 2003).

Information Extraction Tools

There are number of information extraction (IE) technique implementations available as a result of work done in the information extraction and natural language processing communities. Semantic annotation platforms usually take advantage of the work that in some cases has a long history. The most common toolkits are GATE (University of Sheffield, 2004) and Amilcare (University of Sheffield, 2002). GATE is produced by the University of Sheffield's Natural Language Processing Group and is a language process environment for building human language processing systems. GATE is divided into three parts: (1) an architecture which defines a goal component for componentizing a language processing system; (2) a concrete framework implementation; and (3) a graphical development environment. GATE includes an information extraction system called ANNIE (A Nearly-New IE system) that is composed of a tokenizer, gazetteer, sentence splitter, part-of-speech tagger, semantic tagger, coreferencing (OrthoMatcher), and other related IE components.

Amilcare is produced by the University of Sheffield's Computer Science Department and is specifically designed to perform semantic annotation (University of Sheffield, 2002). Amilcare is a rule-based system, where the rules are induced with a learning algorithm run against a training corpus pre-annotated with XML tags. The transformation-based rule algorithm is called LP^2 which induces two types of rules: 1) initial rules for annotating text, and 2) rules that correct mistakes generated by the first type of rules. Interestingly, GATE is used within Amilcare for tokenization, sentence identification, part of speech tagging, gazetteer lookup and named entity recognition.

Initial Ontology

Semantic annotation requires the use of an ontology in order to perform concept instance mapping. Ontologies are usually architected using levels, such as upper and lower. The upper ontology consists of general concepts, while the lower ontology has a deeper specialization of the upper ontology concepts (Missikoff, Navigli, & Velardi, 2002). Some semantic annotation platforms place the responsibility on the user for constructing an initial ontology. Examples include MUSE (Maynard, 2003) and Ont-O-Mat (Handschuh et al., 2002). Other platforms provide an initial ontology as part of their development. The KIM platform provides an ontology called KIMO that is designed to provide a minimal open-domain ontology, and is based on OpenCyc (OpenCyc.org, 2005), WordNet (Princeton University, Cognitive Science Laboratory, 2005), DOLCE (Italian National Resource Council - Institute of Cognitive Science and Technology, 2005) and other upper-level resources (Popov et al., 2003). KIMO is composed of approximately 250 classes, 100 attributes and relations, and the specialization of classes is derived from an analysis of a corpus of general news (Popov et al., 2003). The Seeker component of SemTag uses TAP, which is a shallow knowledge base that contains information about a broad range of popular culture subjects, such as movies, sports, and so forth (Dill et al., 2003). The TAP knowledge base has about 72,000 labels that are used to tag instances found in documents. The MnM platform uses a hand-crafted ontology called KMi (Knowledge Management Institute) (Vargas-Vera et al., 2002). The AeroDAML platform uses the commercial product Aerotext, and utilizes an upper-level ontology based on Wordnet, while the lower-level ontology uses the common knowledge base of AeroText (Kogut & Holmes, 2001). Armadillo provides an example of a platform where the initial ontology is very lightweight, consisting of an address-book type of ontology where members of a computer science department are discovered and populate address information, such as name, phone number, address, and so forth (Dingli et al., 2003).

Table 2 shows some key characteristics for each platform. In this section, the key characteristics among SAPs were compared and investigated to show a sense of each SAP's strengths and weaknesses. With semantic annotation platforms shown in Table 2, we can see the advantages of the SAP architecture in Figure 1 to integrate new external sources or new information extraction tools and how the existing SAPs utilized such an architecture. For example, the Ont-O-Mat system (Handschuh et al., 2002) was originally designed using Amilcare to learn rules about linguistic patterns in order to identify entities. Later the Ont-O-Mat platform was extended using the PANKOW (Pattern-based Annotation through Knowledge on the Web) algorithm (Cimiano et al., 2004). SAPs designed with extensible architectures can adapt to evolving technology. Information extraction components can be replaced as different approaches are developed. The most common toolkits used for entity identification are GATE (University of Sheffield, 2004) and Amilcare (University of Sheffield, 2002).

Future Trends

Semantic annotation platforms for the Web have only recently been developed, and they are not complete in their accuracy and elimination of manual effort. The precision and recall still vary widely depending on the platform used, information extraction (IE) methods, and data source type (unstructured, semistructured, or structured), as shown in Table 2. There still exists an opportunity to improve the performance of SAPs and reduce their required manual effort. A future trend in semantic annotation platforms, then, is the continued integration of technologies originally developed in the field of information extraction.

With the number of available semantic annotation platforms currently available, as shown in Table 2, it is possible to extend existing SAPs with newer annotation implementations that may lead to improved annotation accuracy beyond what current platforms are producing. As an example, the integration of other methods into an existing SAP has been done. The Ont-O-Mat system (Handschuh et al., 2002) was originally designed using Amilcare to learn rules about linguistic patterns in order to identify entities. Further work done by separate researchers developed a method called PANKOW (Pattern-based Annotation through Knowledge on the Web) and integrated it into the Ont-O-Mat platform (Cimiano et al., 2004). The advantage of the integration is that the work could focus on the method rather than on constructing supporting services provided by a SAP. In particular, PANKOW utilizes the ontology and document management facilities provided by Ont-O-Mat (Cimiano et al., 2004).

Conclusion

The Semantic Web requires the widespread availability of document annotations in order to be realized. Benefits of adding meaning to the Web include: query processing using concept-searching rather than keyword-searching (Berners-Lee et al., 2001); custom Web page generation for the visually-impaired (Yesilada et al., 2004); using information in different contexts, depending on the needs and viewpoint of the user (Dill et al., 2003); and question-answering (Kogut & Holmes, 2001). Annotations are currently problematic for several reasons. Manual annotation does not scale to the volume of documents on the Web, and suffers from problems such as annotator motivation and domain knowledge (Bayerl et al., 2003). There are additional problems with manual annotation, such as changing ontologies, and having a document annotated using multiple ontologies, providing multiple perspectives (Dill et al., 2003).

Semantic Annotation Platforms (SAPs) were developed to provide a level of automation to the semantic labeling process, and overcome the limitations of manual annotation. Several semantic annotation platforms currently exist, distinguished primarily by their information extraction method, as that component has the largest impact on the effectiveness of semantic annotation. The two primary approaches are pattern-based and machine learning-based. Machine learning algorithms often perform more effectively than pattern-based methods, but the MUSE system shows that a rule-based system using conditional processing can perform as well as a machine learning system (Maynard, 2003). SAPs designed with extensible architectures can adapt to evolving technology. Information extraction components can be replaced as different approaches are developed. The most common toolkits used within SAPs are GATE (University of Sheffield, 2004) and Amilcare (University of Sheffield, 2002).

Future work on semantic annotation platforms for text-based Web documents will likely focus on integrating other entity identification approaches from the information extraction and natural language processing communities. In addition, ways to bootstrap the semantic annotation process and reduce the amount of manual effort will also evolve. This includes ontology learning and knowledge base population. Extensible platforms will enable the rapid development of new advances by allowing individual components of a semantic annotation platform to be replaced or extended. The continuing evolution of semantic annotation platforms to provide better annotation and new features while extending existing ones is vital to the realization of the Semantic Web.

References

Bayerl, P. S., Lüngen, H., Gut, U., & Paul, K. I. (2003). Methodology for reliable schema development and evaluation of manual annotations. *Proceedings of the Workshop on Knowledge Markup and Semantic Annotation at the Second International Conference on Knowledge Capture (K-CAP 2003),* Florida.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American, 284*(5), 34-43.

Berners-Lee, T. (1998). *Semantic Web road map.* Retrieved April 24, 2005, from http://www.w3.org/DesignIssues/Semantic.html

Brin, S. (1998). Extracting patterns and relations from the World Wide Web. *Proceedings of the WebDB Workshop at 6th International Conference on Extending Database Technology,* Valencia, Spain (Vol. 1590, pp. 172-183).

Chandrasekaran, B., Josephson, J. R., & Benjamins, V. R. (1999). What are ontologies and why do we need them? *IEEE Intelligent Systems, 14*(1), 20-26.

Cimiano, P., Handschuh, S., & Staab, S. (2004). Towards the self-annotating Web. *The 13th International Conference on World Wide Web,* New York (pp. 462-471).

Ciravegna, F. (2001). Adaptive information extraction from text by rule induction and generalisation. *Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI 2001),* Seattle, Washington (pp. 1251-1256).

Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). *GATE: A framework and graphical development environment for robust NLP tools and applications.* The 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02).

Cunningham, H., Maynard, D., & Tablan, V. (2000). *JAPE: A Java annotation patterns engine (2nd Edition).* Technical report CS—00—10, University of Sheffield, Department of Computer Science. Retrieved April 24, 2005, from ftp://ftp.dcs.shef.ac.uk/home/hamish/auto_papers/Cun00e.ps.gz

Cutting, D. (2004). *Apache Jakarta Lucene.* Retrieved April 24, 2005, from http://lucene.apache.org/java/docs/index.html

DARPA (2005). *DARPA agent markup language.* Retrieved June 6, 2005, from http://www.daml.org

Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., et al. (2003). SemTag and seeker: Bootstrapping the Semantic Web via automated semantic annotation. *The 12th International World Wide Web Conference*, Budapest, Hungary (pp. 178-186).

Dingli, A., Ciravegna, F., & Wilks, Y. (2003). Automatic semantic annotation using unsupervised information extraction and integration. *Proceedings of the Workshop on Knowledge Markup and Semantic Annotation at the Second International Conference on Knowledge Capture (K-CAP 2003),* Florida.

Handschuh, S., Staab, S., & Ciravegna, F. (2002). S-CREAM: Semi-automatic CREAtion of metadata. *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02).*

Hearst, M. (1992). Automatic acquisition of hyponyms from large text corpora. *Proceedings of the 14th International Conference on Computational Linguistics,* Nantes, France (Vol. 2, pp. 539-545).

Italian National Resource Council - Institute of Cognitive Science and Technology. (2005). *DOLCE: A descriptive ontology for linguistic and cognitive engineering.* Retrieved April 24, 2005, from http://www.loa-cnr.it/DOLCE.html

Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., Korilov, A., & Goranov, M. (2003). Semantic annotation, indexing, and retrieval. *ISWC 2003, 2nd International Semantic Web Conference,* Sanibel Island, Florida (pp. 834-849).

Kogut, P., & Holmes, W. (2001). AeroDAML: Applying information extraction to generate DAML annotations from Web pages. *Proceedings of the Workshop on Knowledge Markup and Semantic Annotation at the First International Conference on Knowledge Capture (K-CAP 2001),* Victoria, BC.

Kosala, R., & Blockeel, H. (2000). Web mining research: A survey. *SIGKDD Explorations, 2*(1), 1-15.

Kushmerick, N., Weld, D. S., & Doorenbos, R. (1997). Wrapper induction for information extraction. *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI '97),* Nagoya, Japan (pp. 729-737).

Kushmerick, N. (1997). *Wrapper induction for information extraction.* Doctoral dissertation, University of Washington.

Maedche, A., & Staab, S. (2001). Ontology learning for the Semantic Web. *IEEE Intelligent Systems, 16*(2), 72-79.

Lockheed Martin (2005). AeroText. Retrieved June 6, 2005, from http://www.lockheedmartin.com/wms/findPage.do?dsp=fec&ci=11255&rsbci=0&fti=126&ti=0&sc=400

Maynard, D. (2003). Multi-source and multilingual information extraction. *Expert Update.*

Missikoff, M., Navigli, R., & Velardi, P. (2002). The usable ontology: An environment for building and assessing a domain ontology. *The 1st International Semantic Web Conference (ISWC2002)* (pp. 39-53).

Mukherjee, S., Yang, G., & Ramakrishnan, I. V. (2003). Automatic annotation of content-rich HTML documents: Structural and semantic analysis. *Proceedings of the 2nd International Semantic Web Conference,* Sanibel Island, Florida.

OpenCyc.org (2005). *OpenCyc.* Retrieved April 24, 2005, from http://www.opencyc.org/

openRDF.org (2005). Sesame RDF repository. Retrieved June 6, 2005, from http://www.openrdf.org/

Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., & Goranov, M. (2003). KIM: Semantic annotation platform. *The 2nd International Semantic Web Conference (ISWC2003)* (Vol. 2870, pp. 834-849).

Princeton University, Cognitive Science Laboratory. (2005). *WordNet: A lexical database for the English language.* Retrieved April 24, 2005, from http://wordnet.princeton.edu/

Rabiner, L. R. (1989). A tutorial on hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), 257-285.

Reeve, L., & Han, H. (2005). Survey of semantic annotation platforms. *Proceedings of the 20th Annual ACM Symposium on Applied Computing, Web Technologies and Applications track,* Santa Fe, New Mexico.

Ondrej, S., Labsky, M., & Svatek, V. (2004). RDF-based retrieval of information extracted from Web product catalogues. *Proceedings of the 27th Annual ACM SIGIR Conference on Research and Development in Information Retrieval, Semantic Web Workshop,* Sheffield, UK,

Tallis, M. (2003). Semantic word processing for content authors. *The 2nd International Conference on Knowledge Capture,* Sanibel, Florida.

University of Sheffield (2004). *GATE - A general architecture for text engineering.* Retrieved December 28, 2004, from http://gate.ac.uk/

University of Sheffield (2002). *Amilcare – Adaptive IE tool.* Retrieved December 28, 2004, from http://nlp.shef.ac.uk/amilcare/amilcare.html

Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A., & Ciravegna, F. (2002). MnM: Ontology driven semi-automatic and automatic support for semantic markup. *The 13th International Conference on Knowledge Engineering and Management (EKAW 2002)* (pp. 379-391).

Yesilada, Y., Harper, S., Goble, C., & Stevens, R. (2004). Ontology based semantic annotation for visually impaired Web travellers. *Proceedings of the 4th International Conference on Web Engineering (ICWE 2004)*, Munich, Germany (Vol. 3140, pp, 445-458).

Section III

Ontologies-Based Querying and Knowledge Discovery