Integrating Hidden Markov Models Into Semantic Web Annotation Platforms

Larry Reeve
for Dr. Weber
INFO629, Fall 2004
Term Project

<u>Table of Contents</u>

Semantic Annotation Platform Overview…………………………………………… 1

Information Extraction Approaches Used in Semantic Annotation Platforms…………… 4

Hidden Markov Models………………………………………………………… 6

Hidden Markov Models and Semantic Annotation…………………………..…. 8

Integration of HMMs into Semantic Annotation Platforms….…………………..…..…. 11

References……………………………………………………………………… 14

Semantic Annotation Platform Overview

The Semantic Web, as proposed by Tim Berners-Lee (1998), is the next generation of the Web that is focused not on presentation elements of the Web, but on bringing structure to Web-based information in order to provide machine-processable 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, Hendler, & Lassila, 2001). 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).

A key problem with the Semantic Web is providing document markup, or annotations, for both existing and new documents on the Web. Annotations are what provide the link between information stored within a document and the ontology (Berners-Lee et al., 2001). Semantic annotation is to tag ontology class instance data and map it into ontology classes. Semantic annotation can be seen as a typical information extraction 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 (Popov et al., 2003). Semantic annotation has been developed using research done in the areas of information extraction, information integration, wrapper induction, and machine learning (Dingli, Ciravegna, & Wilks, 2003). Manual approaches were the initial approaches to annotation because automatic or semi-automatic systems were not yet available. Manual annotation can be done using tools such as Semantic Word (Tallis, 2003), which provides an environment for authoring as well as marking up documents from within a single interface. Manual approaches suffer from several drawbacks. It is expensive to have human annotators markup documents (Cimiano, Handschuh, & Staab, 2004). A human annotator may not consider using multiple ontologies (Dingli et al., 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 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).

To overcome the limitations of manual annotation, semi-automatic systems were developed. Semi-automatic systems, rather than completely automatic systems, are used because it is not yet possible to automatically identify and classify all entities within source documents with complete accuracy (Popov et al., 2003). Building a completely automatic annotation system is an open research problem. Some advantages of semi-automatic annotation are obvious, such as providing document volume scalability by reducing or potentially eliminating the human workload (Dill et al., 2003), but other reasons for reducing human involvement are not so obvious. For example, human annotators can provide unreliable annotation, and for many reasons: complex ontology schemas, unfamiliarity with subject material, and motivation, to name a few (Bayerl, Lüngen, Gut, & Paul, 2003)

Semi-automatic annotation systems can be viewed as platforms, or infrastructure, for providing semantic annotation services (Popov et al., 2003). These semantic annotation platforms (SAP) are composed of components that can often be interchangeably replaced. Figure 1 shows the general architecture of a SAP. The figure shows that semantic annotation platforms are generally architected as composable systems. Most SAPs are extensible, meaning various components can be replaced with alternate implementations. For example, the lower interface contains the critical components for semantic annotation performance. In an extensible SAP, a rule induction 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 (Maynard, 2003).

Examples of existing annotation platforms include AeroDAML (Kogut & Holmes, 2001), Armadillo (Dingli et al., 2003), MnM (Vargas-Vera et al., 2002), MUSE (Maynard, 2003), Ont-O-Mat (Handschuh, Staab, & Ciravogna, 2002), and SemTag/Seeker (Dill et al., 2003). The platforms are primarily distinguished by 1) the features offered, 2) the information extraction method used to find entities within documents, and 3) whether or not they are extensible. Features offered by SAPs include ontology and knowledgebase management (storage, editors), access APIs, annotation storage to allow multiple ontologies/annotations per document, and information extraction methods.

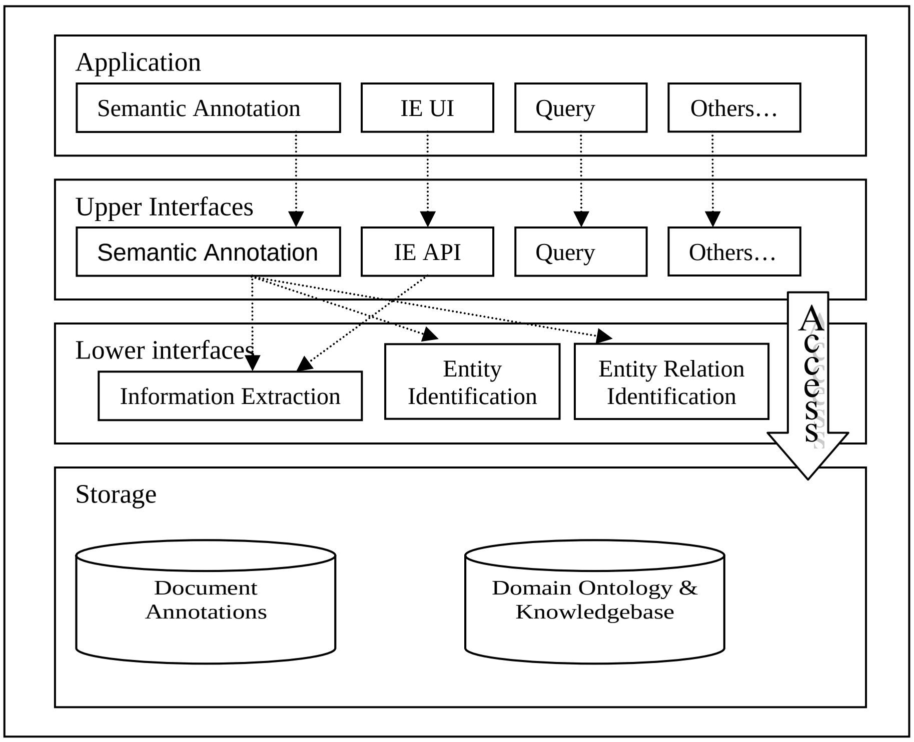

Figure 1: General architecture of a semantic annotation platform.

Information Extraction Approaches Used in Semantic Annotation Platforms

Current semantic annotation platforms use several methods for information extraction (IE) from Web documents: rules, wrappers, and patterns. Table 1 lists several SAPs and the information extraction method used by each platform. Rules are typically hand-crafted rules that define how entities can be found in text (Maynard, 2003). Examples of such systems are AeroDAML (Kogut & Holmes, 2001) and KIM (Popov et al., 2003). 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. Rules can also be learned from inductive machine learning, in order to reduce the effort required to build manual rules. An example of such a system is the SemTag/Seeker platform (Dill et al., 2003). SemTag uses labels stored in the knowledgebase to find instances of ontology concepts, and then uses statistical likelihood to determine where in the ontology tree an instance is most likely to be contained. The process is not completely automatic, however, as an initial set of training data must be provided. The authors report 700 entries as the initial size of the training set (Dill et al., 2003). The MUSE platform is an interesting rule-based system because it can adapt the rules used in entity identification depending on text attributes, such as language, document type, document source, and so forth (Maynard, 2003).

Table 1: Semantic annotation platform information extraction methods and their performance measurements.

Platform	IE Method	Precision	Recall
AeroDAML	Manual Rules	n/a	n/a
Armadillo	Pattern Discovery	91	74
KIM	Manual Rules	86	82
MnM	Wrapper Induction	95	90
MUSE	Manual Rules	93	92
Ont-O-Mat using Amilcare	Wrapper Induction	n/a	n/a
Ont-O-Mat using PANKOW	Pattern Discovery	65	28
SemTag	Semi-automatic Rules	82	n/a

Wrappers are another method of information extraction used in semantic annotation platforms. Kushmerick (1997) defines wrappers as a mapping from a page to a set of relational 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). Examples of SAPs using wrappers are MnM and Ont-O-Mat using Amilcare.

Patterns are also a widely-used technique 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, to find entities (Cimiano et al., 2004). The Armadillo system is an example of a pattern-based platform. The Ont-O-Mat platform has also been updated to replace the wrapper IE component with a pattern-based component, called PANKOW, and also demonstrates the usefulness of an extensible SAP, where components can be replaced without losing or duplicating the features already available within an SAP (Cimiano et al., 2004).

SAPs can take advantage of existing information extraction frameworks. For example, the GATE framework (Cunningham, Maynard, Bontcheva, & Tablan, 2002) contains many pre-built components for performing IE, such as tokenizers, part-of-speech taggers and gazetteers. Amilcare (Handschuh et al., 2002) is another example, where the framework is used to induce linguistic rules, based on the LP^2 algorithm from the natural language processing community.

Hidden Markov Models

Semantic annotation platforms for the Web have only recently been developed, and they are not complete in their accuracy and manual effort required. The precision and recall still vary widely depending on the platform used, IE methods, and data source type (unstructured, semi-structured, or structured), as shown in Table 1. There is still exists opportunity to improve the performance of SAPs and reduce their required effort. A promising approach is the use of Hidden Markov Models.

Hidden Markov Models (HMMs) are models of output produced by real-world processes (Rabiner, 1989). Examples include acoustic signals used to identify speech, and bibliographic entries used to identify published works. HMMs are distinguished by the fact that they can only observe outputs, and not the events that produced the output. The models are then generated using only the outputs, in an attempt to understand the events which produced them (Rabiner, 1989). Rabiner (1989) describes a simple example of this. Assume there is a set of coin-toss using two coins. The person tossing coins is located behind a wall so that you cannot see the coin toss occur. As each coin toss occurs, the person calls out heads or tails. The modeling task is to explain the sequence of observed heads or tails.

A more practical example involves the use of HMMs in the AutoBib system (Geng & Yang, 2004). AutoBib is designed to extract information from bibliographic sources. The idea is to take the regularity of a bibliographic citation, and extract various fields, such as Title, Author, Publisher, and Year. In the AutoBib application, the observations are the instances of each defined field (for example, 'Yang' is the 'Author'), and the hidden states are the fields (no values) Title, Author, Year, and so forth. Bibliographic information is an example of a structure modeled by an HMM.

HMMs are constructed using finite state machines. The state transitions are associated with a probability on each link, indicating the probability of transition from one state to the next. Each state also has an associated set of output probabilities. The output probabilities define what the probability is for each of a set of defined output symbols (Duda, Hart, & Stork, 2000). A core challenge is defining what the states are, since that is the basis of constructing a model (Rabiner, 1989).

A problem with HMMs is that they require a set of data to train the model with. Once the states of the model have been defined, the transition probabilities and the output probabilities associated with each state need to be trained. Training normally occurs by setting the transition and output probability matrix values to some arbitrary starting value, and then iteratively refining the probability values in each matrix using the training data (Duda et al., 2000). The amount of manual effort required to produce the training set can be reduced by providing a method to bootstrap the system with data, and then automatically expanding the number of training samples (Geng & Yang, 2004). As an example of the effort required to train HMMs, the MUSE system reports that their rule-based system, based on manual rule generation, can be ported to different languages faster than machine-learning based systems because of the effort required to generate training samples (Maynard, 2003).

Hidden Markov Models have been used successfully in the Information Extraction field for parsing structured information, such as addresses (Borkar, Deshmukhy, & Sarawagiz, 2001), and bibliographic information (Geng & Yang, 2004). A key attribute of HMMs is that they are adaptable in the case where information may be missing. For example, in structured record extraction, HMMs can handle variations in the record structure, such as missing or optional fields (Geng & Yang, 2004). A visualization of a typical HMM is shown in Figure 2.

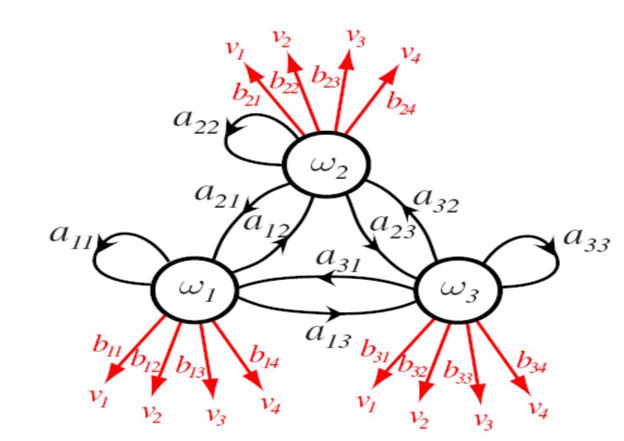

Figure 2: Visualization of typical HMM. w's are hidden states, a's are state transition probabilities, and b's are observation probabilities (Rabiner, 1989)

Hidden Markov Models and Semantic Annotation

Applying Hidden Markov Models (HMMs) to semantic annotation has just recently begun to be explored. The DATAMOLD tool (Borkar et al., 2001) was designed to extract address information from various datasets as part of the cleansing process in building a data warehouse. The problem DATAMOLD was designed to solve is to extract information from different text styles, including free-form text. This is a similar process to semantic annotation, where documents may be in structured, unstructured, or semi-structured form, and the goal is to identify entities within the text and assign them to slots in the ontology. The AutoBib system (Geng & Yang, 2004) uses HMMs in a similar way, except the application domain is bibliographic information. Also, AutoBib provides a facility for automatically generating training samples from a few seed samples. Both DATAMOLD and AutoBib represent the text within the model as a sequence of words. The words form the set of output symbols that a particular state may observe. This approach has led to good results for DATAMOLD, depending on the domain of addresses. For U.S. addresses, overall precision and recall is 99.605%, while for corporate addresses, overall precision and recall is 83.656% (Borkar et al., 2001). AutoBib saw similar results, depending on the data source type. For the DBLP bibliographic data source, the measured accuracy is 98.9%, while for the CSWD data source 93.4% accuracy was obtained (Geng & Yang, 2004).

In an effort to improve effectiveness with complex language texts, such as biomedical texts, work was done to construct HMMs by incorporating grammatical structure information. The idea is to perform a shallow parse of a sentence in order to generate phrase segments of a particular type, such as noun-phrase and verb-phrase. These phrases are then labeled if they contain words from the target terminology. The labeled phrases are then used as the hidden states within the HMM (Ray & Craven, 2001). The result is that this phrase model works better for some types of datasets as compared to the sequence-of-words model, while it can actually be lower performing for other types of datasets (Ray & Craven, 2001). The architecture of this model is shown in Figure 3. The same group later did expanded work to build Hierarchical Hidden Markov models, in which there are two layers of an HMM rather than one. The upper level represents phrases, as before, but the lower level represents individual words within a phrase (Skounakis, Craven, & Ray, 2003). This model results in improved performance for all tested data sets, as compared to the sequences of words model (Skounakis et al., 2003). The architecture of the phrase model is shown in Figure 4.

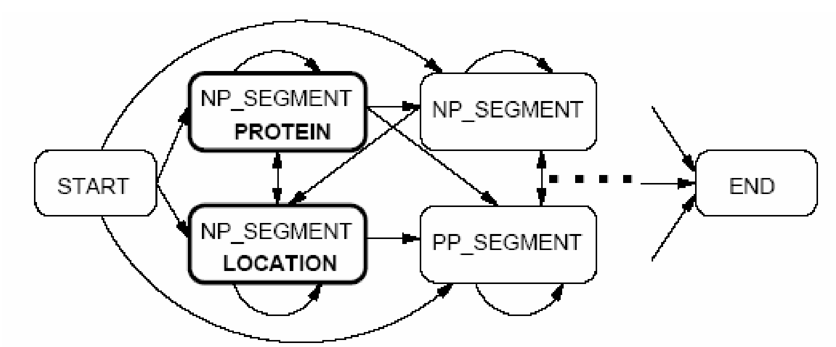

Figure 3: Architecture of a grammatical-information HMM based on part-of-speech phrases. (Ray & Craven, 2001)

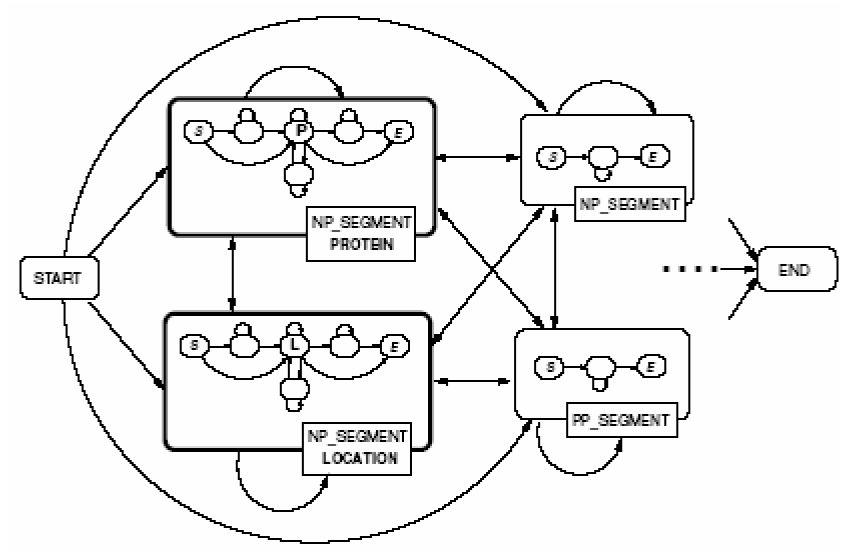

Figure 4: Architecture of a hierarchical phrase-based HMM. The dark rectangles indicate the upper level HMM, while the contents of the rectangle are the lower level HMM. (Skounakis et al., 2003)

All of these efforts are stepping stones to integrate HMMs in order to perform semantic annotation. Most of the focus on HMMs has been on producing information extraction methods, rather than integrating the methods into a larger semantic annotation platform. There has been some recent effort to use HMMs for Semantic Web annotation, although none to date has integrated the HMM methods from the IE field into a semantic annotation platform. The Rainbow project (Svab Ondrej, Labsky Martin, Svatek Vojtech, 2004), which aims to build a domain-specific semantic search engine, has implemented HMMs in order to perform semantic annotation. The implemented HMM uses the sequence-of-words model. The target documents are HTML pages, and each page has transformation applied before the HMM is used. The transformations include: 1) HTML block elements (such as table cells) are identified; 2) inline HTML element strings are converted into abstract tags (such as <important>); and 3) manual rules are applied to replace well-known sections of forms with tags (Svab Ondrej, Labsky Martin, Svatek Vojtech, 2004). This means that substantial pre-processing must occur before the HMM can be applied, and some of this pre-processing is manually generated. An interesting aspect of the Rainbow project HMM is that a single model incorporating all semantic slots to be filled is used, rather than a model (HMM) for each semantic slot (Svab Ondrej, Labsky Martin, Svatek Vojtech, 2004). The ontology used defines product catalog information for bicycle products, so the domain is very narrow and the ontology slots to be filled number only a few. The results of the HMM application are different depending on the semantic slot. The best slot, *price*, has a precision of 98.9% and a recall of 89.5%, while the worst performing slot, *picture*, has a precision of 89.6% with a recall of 69% (Svab Ondrej, Labsky Martin, Svatek Vojtech, 2004). Further work to extract sets of slots, in order to perform template extraction of an entire ontology class rather than single slot extraction of each ontology class property, results in a less than 50% match as compared to manual annotation (Svab Ondrej, Labsky Martin, Svatek Vojtech, 2004). The result is that single slot (unary) extraction is currently more effective with their current work than multi-slot (n-ary) extraction. Project Rainbow expects to provide some level of annotation services similar to what is available in a semantic annotation platform in future work. However, there are no plans to integrate its HMM work into an existing SAP. The CROSSMARC project also looks to integrate HMMs into the semantic annotation process. CROSSMARC focuses on annotating named entities, and uses the HMM to find new entities not found by its current semantic tagger, which uses string matching, similar to a gazetteer (Valarakos, Sigletos, Karkaletsis, & Paliouras, 2003). The idea is to supplement the string matching to find known entities, and use the HMM to find additional entities that are not currently known. Once new entities are discovered, they are added to the ontology and become known entities. In this way, the pool of known entities continues to grow for subsequent annotation processing.

Integration of HMMs into Semantic Annotation Platforms

With the number of available semantic annotation platforms currently available, as shown in Table 1, it is possible to extend existing SAPs with newer annotation implementations that may lead to improved annotation accuracy beyond what current platforms are producing. While work has been done to use HMMs in information extraction, as discussed previously, and related work has also been done to perform annotation in a domain search project, HMMs have not been integrated with any of the SAPs to date. It seems reasonable that this prior work could be used as a starting point to integrate HMMs into one of the SAPs.

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).

A primary disadvantage HMMs have over the rule-based systems that are largely implemented in current SAPs (Svab Ondrej, Labsky Martin, Svatek Vojtech, 2004) is the training problem. The data required to get good accuracy out of an HMM may be fairly large (Duda et al., 2000). If this training data must be manually produced, a bottleneck and maintenance issue results. However, since SAPs normally have a knowledgebase as a component of the system, it may be possible to develop a way to bootstrap the training process using the knowledgebase (Valarakos et al., 2003). For example, in the CROSSMARC project, entities are tagged using information from the ontology and knowledgebase (Valarakos et al., 2003). These annotations are then fed into a HMM in order to train it. The HMM is then used to find additional entities not tagged by the ontology tagger.

As discussed, the integration of HMMs into semantic annotation platforms is an uncompleted effort. HMMs offer advantages over existing semantic annotation approaches, such as rules. For example, HMMs are more flexible than static rules due to their probabilistic nature. HMMs also have disadvantages, primarily the amount of training data required. However, this disadvantage may reduced or even eliminated by using information contain in the SAP ontology and knowledgebase to bootstrap annotated examples. However, current HMM work developed outside of SAPs are not able to take advantage of such platform features. It also anticipated that HMMs focusing directly on semantic annotation may also be developed, such as the HMM work the Rainbow project started (Skounakis et al., 2003). Semantic HMMs would focus on filling semantic slots directly from text rather than going through a HMM at the information extraction stage. Typical information extraction HMMs can locate entities, which are then semantically tagged using gazetteers or other semantic tagging methods (Skounakis et al., 2003).

Semantic annotation platforms, then, would benefit from HMMs by taking advantage of potentially improved annotation accuracy through their inherent structure flexibility over alternate methods, such as rules. At the same time, SAPs provide supporting features that can potentially mitigate the machine learning training disadvantage of HMMs. The continuing evolution of SAPs to provide better annotation is vital to the realization of the Semantic Web, and the integration of Hidden Markov Model approaches should be part of the SAP evolution.

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, USA.

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

Borkar, V., Deshmukhy, K., & Sarawagiz, S. (2001). Automatic Segmentation of Text into Structured Records. *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data,* California, USA, *30*(2) 175-186.

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.*

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. *Thirteenth International Conference on World Wide Web,* New York, NY, USA, 462-471.

Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02).*

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. *Twelfth International World Wide Web Conference,* Budapest, Hungary, 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, USA.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classification* (2nd ed.). Hoboken, NJ, USA: Wiley Interscience.

Geng, J., & Yang, J. (2004). Automatic Extraction of Bibliographic Information on the Web. *Proceedings of the 8th International Database Engineering and Applications Symposium (IDEAS'04),* Coimbra, Portugal, *00* 193-204.

Handschuh, S., Staab, S., & Ciravogna, F. (2002). S-CREAM -- Semi-automatic CREAtion of Metadata. *SAAKM 2002 -Semantic Authoring, Annotation & Knowledge Markup - Preliminary Workshop Programme.*

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.

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

Maynard, D. (2003). Multi-Source and Multilingual Information Extraction. *Expert Update,*

Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., & Goranov, M. (2003). KIM - Semantic Annotation Platform. *2nd International Semantic Web Conference (ISWC2003), 2870* 834-849.

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE, 77*(2), 257-285.

Ray, S., & Craven, M. (2001). Representing Sentence Structure in Hidden Markov Models for Information Extraction. *Proceedings of the 17th International Joint Conference on Artificial Intelligence,* Seattle, WA, 1273-1279.

Skounakis, M., Craven, M., & Ray, S. (2003). Hierarchical Hidden Markov Models for Information Extraction. *Proceedings of the 18th International Joint Conference on Artificial Intelligence,* Acapulco, Mexico, 427-433.

Svab Ondrej, Labsky Martin, Svatek Vojtech. (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. *Second International Conference on Knowledge Capture,* Florida, USA.

Valarakos, A., Sigletos, G., Karkaletsis, V., & Paliouras, G. (2003). A Methodology for Semantically Annotating a Corpus Using a Domain Ontology and Machine Learning. *Proceedings of the RANLP'2003 Conference,* Borovets, Bulgaria, 495-499.

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),* 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, *3140* 445-458.