A Survey of Domain Ontology Engineering: Methods and Tools

Amal Zouaq

Simon Fraser University, School of Interactive Arts and Technology, 13450 102 Ave. Surrey, BC V3T 5X3, Canada, email: amal.zouaq@athabascau.ca, azouaq@sfu.ca

Abstract With the advent of the Semantic Web, the field of domain ontology engineering has gained more and more importance. This innovative field may have a big impact on computer-based education and will certainly contribute to its development. This paper presents a survey on domain ontology engineering and especially domain ontology learning. The paper focuses particularly on automatic methods for ontology learning from texts. It summarizes the state of the art in natural language processing techniques and statistical and machine learning techniques for ontology extraction. It also explains how intelligent tutoring systems may benefit from this engineering and talks about the challenges that face the field.

1. Introduction

Intelligent Tutoring Systems (ITSs) are computer-based instructional systems that are composed of four principal modules: the expert module, the learner module, the tutor module and the interface. The expert module is in charge of the learning content which indicates what can be taught by the ITS (the domain model). In this regard, one of the most acute research issues is how the expert module can be effectively modeled, what kind of knowledge representations are available and what kind of knowledge acquisition techniques are applicable. In fact, one of the main obstacles to ITSs development and wide dissemination is the cost of their knowledge base and particularly the cost of producing the domain model from scratch. In front of this knowledge acquisition bottleneck, many attempts have been made to create automated methods for domain knowledge creation. However, these attempts have not been as successful as we could wish. Moreover, these efforts have not led to reusable and standard methods and formalisms for knowledge base creation and update.

The advent of the Semantic Web has created new research avenues, especially with the domain ontology engineering field. The necessity of creating domain ontologies in a (semi) automatic way has been quickly understood by the research community. Using domain ontologies for knowledge representation has two main advantages: first, their standard formalism enables sharing and reusing ontologies between any ontology-friendly environments. Second, their formal structure enables to reason over the obtained knowledge representations and to envisage the automatic extraction of the ontological components as modular layers. This automatic ontological extraction is known as "**Ontology Learning**".

From a computer-based education perspective, domain ontologies can be considered as a way to bridge the gap between eLearning and ITS by providing common and sharable resources, standard representations and common reasoning mechanisms. They can also provide a way to build domain knowledge bases that can be widely disseminated and that can be reused and updated with semi-automatic methods.

This paper presents an overview of domain ontology engineering and focuses particularly on automatic methods for ontology learning, especially from texts. It is organized as follows. After the introduction, section 2 introduces the state-of-the art and the history of knowledge representations within ITS. Section 3 explains the field of ontology engineering and provides an overview of the ontology learning process from text. Each task and component of this process is explained. We also present, for each task, the natural language processing (NLP) techniques and the statistical and machine learning techniques. Section four and five introduce very briefly the ontology learning process from other sources than text as well as the ontology update task. Section six indicates the challenges that face domain ontology engineering in general as well as more specific ITS-related challenges. Finally, a conclusion summarizes the whole paper.

2. Retrospective

ITSs have their root in expert systems and one of their distinctive feature is that their knowledge base is not composed of a set of static frames (such as those of traditional Computer-Assisted Instruction) but integrates a rich knowledge structure on which the ITS is able to reason. Traditionally, domain knowledge has been represented by black-box domain models and by glass-box domain models (Polson and Richardson, 88). Black-box domain models organize domain knowledge and provide accurate solutions to learners but they cannot provide detailed explanations about their reasoning. On the opposite, glass-box domain models are able to explain their inferences step by step. In both kinds of domain models, there is a need to represent declarative knowledge and procedural knowledge. **Declarative knowledge** represents factual and conceptual information while **procedural knowledge** represents action sequences and problem solving procedures that are normally followed by a domain expert. The main objective behind these representations is to enable the system to possess a structured knowledge, to facilitate reasoning and to be able to provide the best adapted learning session to a learner.

Much research has been devoted to domain modeling in intelligent tutoring systems for declarative domain knowledge and for procedural domain knowledge. Various representation formalisms have been proposed and used in intelligent tutoring systems such as rules (Vassileva, 98), semantic networks (Hsieh et al., 99), Bayesian networks (Van Lehn and Dong, 2001), case-based reasoning (Guin-Duclosson et al., 2002) and fuzzy logic (Nkambou, 99). Based on (Hatzilygeroudis and Prentzas, 2004), this section summarizes the most important techniques used for knowledge representation within ITS.

Semantic networks and other related knowledge representations (concept maps, topic maps, conceptual graphs, etc.) represent knowledge in the form of a graph composed of nodes (concepts) and links between nodes, which model the conceptual relationships between the concepts. One type of semantic networks that is interesting for intelligent tutoring systems is the concept map. Intelligent tutoring systems can benefit from domain concept maps for their domain model and their learner model (Kumar 2006). In their philosophy, concept maps are very close to the topic map approach (Garshol, 2004) as the concepts can be considered as "topics" with occurrences and associations. This is especially interesting for indexing learning content with concepts and associations and for information retrieval purposes. Another example of representations is the conceptual graphs. Based on existential graphs and semantic networks, conceptual graphs enable a direct transformation of natural language into a logical representation. (Kabbaj et al., 96) have used Synergy, a language based on conceptual graphs, to simulate the generation of a course from a curriculum.

Procedural knowledge has been best described in expert systems using heuristic rules. These kinds of rules are intuitive and easy to understand by a domain expert. However, with the growing number of rules, the inference process may not be efficient. Knowledge acquisition is another bottleneck of the approach, since it is very difficult to learn this kind of expertise without relying too heavily on domain experts. Finally, rules necessitate a careful planning of the input and cannot handle unknown or incomplete inputs. Case-based reasoning (CBR) can represent a solution to the problem of handling unknown inputs. CBR relies on a case knowledge base which is used to draw the most appropriate conclusion whenever a new case is presented. One of the strengths of this approach is that it is capable of providing a conclusion even for unknown cases due to a similarity function. It is also an intuitive technique that can be more easily developed compared to other formalisms. However, it is not able to model heuristic or uncertain knowledge and again, it relies heavily on domain experts for building the initial case base.

In general, all the knowledge acquisition process is a heavy task. This leads to difficulties in building, reusing and propagating intelligent tutoring systems. In fact, there is a need of standard representations that enable the modularization of the intelligent tutoring system creation, evolution and maintenance. There is also a need of explicitly providing semantic relationships between the learning content concepts and of developing pedagogical activities that are built on this domain knowledge. With the advent of the Semantic Web, many questions have been raised about how the domain model could benefit from the Semantic Web languages and techniques. A successful integration of the Semantic Web and the ITS philosophy could enable better reuse of ITS components and better sharing and engineering of domain knowledge. Because ontologies are the backbone of the Semantic Web, using these ontologies to represent domain knowledge and also instructional knowledge is an interesting avenue. These questions have been particularly acute with the raise of the *Educational Semantic Web* (Aroyo and Dicheva, 2004), which comes from the eLearning field and which has proposed the use of ontologies to index and structure the learning content. The intelligent tutoring systems have been slower in their adoption of the ontology concept, especially for modeling domain knowledge but this is now an undeniable fact. Intelligent tutoring systems can benefit from ontology engineering because ontologies represent a standard way for modeling knowledge. They are expressed using formal and standard languages which enable sharing and reasoning. Moreover, there is a growing awareness within the ITS and the eLearning communities of the importance of adopting common methods for domain knowledge acquisition and representation. In this way, ITS will benefit from the huge number of available eLearning resources. Similarly, eLearning systems will benefit from the ITS domain modeling and reasoning. Finally, since ITSs are domain dependant, it is very important to develop easy and reusable knowledge acquisition tools and to integrate automatic methods for this acquisition and evolution. Ontology engineering can provide an answer to these needs and the following section introduces the reader to domain ontology engineering.

3. Domain Ontologies Acquisition

Before going further into detail, an important step is to define the notion of ontology. Very briefly, an ontology is a formal specification of a conceptualization (here a domain) and it includes the definition of classes, objects, properties, relationships and axioms. Ontologies are expressed using a formal language such as RDF or OWL and support automatic inference. Generally, ontologies imply a kind of consensus within a community, meaning that they formalize concepts that are generally accepted within this community. There are many kinds of ontologies such as upper-level ontologies, task ontologies and domain ontologies. Here, we are especially interested in domain ontologies.

3.1. Ontology and Ontology Engineering

As previously said, the concept of a domain ontology as envisioned by the eLearning community is relatively new in the field of ITS. However, domain ontology engineering is a growing research area that has received much attention in other fields and it is the angular stone of the Semantic Web. Ontology engineering is a field that explores the methods and tools for handling the ontology lifecycle. It requires a general and domain-independent methodology that provides guidance for the ontology building, refinement and evaluation (Guarino and Welty, 2002). The ontology life-cycle can be schematized by four main stages: the specification stage, the formalization stage, the maintenance stage, and finally the evaluation stage.

- *The specification stage* allows the identification of the purpose and the scope of the ontology. The specification stage relies generally heavily on domain experts and requires the definition of competency questions that the ontology must be able to answer. It is also dependent on the application that is going to use the ontology;
- *The formalization stage* produces a conceptual and formal model that satisfies the specification stage;
- *The maintenance stage* allows ontology update and evolution, and checks it consistency;
- Finally, the *evaluation stage* analyzes the resulting ontology and checks if it satisfies the initial needs and if it has the desired features.

Here, we are especially interested in the formalization stage and in how this stage can benefit from automated methods for knowledge acquisition. In fact, the most common and successful approaches for domain engineering are generally manual and the best authoring tools or ontology editors can help the expert formalize his knowledge but they are generally very far from an automated procedure. Hence, it is interesting to state explicitly the steps that can be automated in order to alleviate the task of human experts and the burden of knowledge acquisition. Ontology learning techniques have been adopted to reach this goal (Aussenac-gilles et al., 2000). These learning techniques can vary according to the degree of automation (semi-automatic, fully automatic), the ontological knowledge that has to be extracted (concepts, taxonomy, conceptual relationships, attributes, instances, axioms), the knowledge sources (texts, databases, xml documents, etc.) and finally the purpose (creating ontologies from scratch and/or updating existing ontologies).

3.2. Ontology Learning Knowledge Sources

Ontology learning involves the use of some sort of data (structured or semi-structured or unstructured) as input to the learning process. Structured data refer to already defined knowledge models including database schemas or existing ontologies. Semi-structured data designates the use of some mixed structured data with free text such as Web pages, Wikipedia, dictionaries and XML documents. Unstructured data is related to any textual content. Generally, the existence of a structure helps direct the ontology learning process towards relevant parts of data.

Most of the approaches for ontology learning from (semi)structured sources rely on linguistic and statistical techniques and use the underlying schema already available in the structure. For instance, some works rely on dictionaries such as WordNet and try to parse natural language definitions. Examples of such works include OntoLearn (Navigli et al., 2004) (Velardi et al., 2005) and (Rigau et al., 98). Others rely on thesauri as the knowledge source (Van Assem et al., 2004) or on xml schemas (Volz et al., 2003) which are converted into a domain ontology by translating non-terminal and terminal symbols into concepts and roles. Similarly, the work of (Stojanovic et al., 2002) uses a rule mapping scheme in order to convert an xml schema or a relational database schema into a domain ontology. We can also cite the work of (Delteil et al., 2001) which created an ontology learning procedure from RDF annotations and (Nyulas et al., 2007) which developed a plug-in for importing relational databases into an ontology editing environment (Protégé). Rule Knowledge bases have also been used to create an ontology (Suryanto and Compton, 2001) using statistical measures. The work of (Jannink and Wiederhold, 1999) extracts a graph structure from dictionaries and uses statistical filtering and the PageRank algorithm in order to determine important relationships and concepts. Another example is the work of [Papatheodorou et al., 2002], who build taxonomies using cluster mining from xml or RDF domain repositories.

In general, knowing the knowledge source structure helps and guides the ontology learning process. However, the majority of the available electronic data is in the form of unstructured documents. Being able to exploit huge domain corpora and electronic publications is then a requirement for ontology acquisition. This is the reason why this chapter focuses particularly on ontology learning from texts.

4. Building Domain Ontology from Texts

The first step in the ontology learning from text is to prepare a corpus related to the domain of interest (specification stage). This corpus has to be carefully chosen and should describe adequately the domain. A number of sub-tasks have then to be performed in order to learn a domain ontology including concepts, taxonomy, conceptual relationships, attributes, instances and axioms learning. Examples of systems that carry on the whole ontology learning task include Text-2-Onto (Cimiano and Volker, 2005a), TEXCOMON (Zouaq and Nkambou, 2009a) (Zouaq and Nkambou, 2009b), OntoLearn (Velardi et al., 2005), and OntoGen (Fortuna et al., 2007). In the following sections, we highlight state-of-the-art knowledge extraction techniques for each sub-task of ontology learning. In every step, we present the NLP-based approaches and the machine learning and statistical approaches.

4.1. *Concept Extraction*

The first task that has to be performed in ontology engineering is the identification of concepts. Concepts can be described as complex mental objects that are characterized by a number of features. Concept extraction refers to the identification of important domain classes.

In the terminological approaches, concepts are terms that are particularly important for the domain. These terms are generally extracted from the corpus as outlined by (Buitelaar et al., 2005) who consider that a concept should have a linguistic realization. In this case, the major challenge is to be able to differentiate domain terms from non domain terms, usually using statistical filtering. The identified terms (composed from single or several words) can then be either considered as concepts/classes or they can be classified according to broad classes already available in thesauri and vocabularies. Other approaches rely on clustering and machine learning in order to learn semantic classes. In this case, a concept may have no corresponding term in the corpus. This is further explained in the following paragraphs.

4.1.1. NLP-based Techniques

NLP-based techniques for concept learning consider terms as candidate concepts. These approaches rely on linguistic knowledge and use parsers and taggers in order to determine the syntactic roles of terms or to discover linguistic patterns. Typically, some works adopt a surface analysis by running a part-of-speech tagger over the corpus and by identifying manually defined patterns (Sabou, 2005) (Moldovan and Girju, 2001) while others use a deeper analysis and use a NLP parser (Reinberger and Spyns, 2005) (Zouaq and Nkambou, 2009a). In general, the syntactic analysis identifies the nominal phrases that may be important for the domain. For example, (Zouaq and Nkambou, 2009a) use dependency relationships indicating nominal phrases such *nominal subject*, *direct object* and *noun compound modifier* in order to detect these nominal phrases. Most of the time, there is also a list of manually defined seed words that triggers the ontology learning process. However, (Zouaq and Nkambou, 2009a) proposed the use of an automatic keyword extractor in order to automate this task.

4.1.2. Statistical and machine Learning techniques

Usually, NLP-based approaches are not used alone and require statistical filtering. Statistical approaches consider all important terms in a domain as potential concepts and require quantitative metrics to measure the importance of a term. Such quantitative measures include the popular TF*IDF (Salton and Buckley, 88) and C-value/NC-value (Frantzi et al., 98). The employed measures can differ depending on the application.

Based on Harris Distributional hypothesis (Harris, 1954), clustering techniques can also be used in order to induce semantic classes (Almuraheb and Poesio, 2004) (Lin and Pantel 2001). Here a concept is considered as a cluster of related and similar terms. Harris' hypothesis, which is the basis of word space models, states that words that occur in similar contexts often share related meaning (Sahlgren 2006). Term similarity can be computed using collocations (Lin 99), co-occurrences (Widdows and Dorow, 2002) and latent semantic analysis (Hearst and Schutze, 1993). For example, (Lin and Pantel, 2001) represent each word by a feature vector that corresponds to a context in which the word occurs. The features are specific dependency relationships coupled with their occurrence in the corpus. The obtained vectors are then used to calculate the similarity of different terms using measures such as mutual information (Hindle, 90) (Lin, 98) and to create clusters of similar terms. Comparable approaches include Formal concept Analysis (such as the approach presented in (Cimiano, 2006)) and Latent Semantic Indexing algorithms (e.g. Fortuna et al., 2005). These approaches build attributes/values pairs that correspond to concepts.

Statistical approaches can also be used on top of NLP-based approaches in order to identify only relevant domain terms by comparing the distribution of terms between corpora (Navigli and Velardi, 2004). Another approach used by (Velardi et al., 2005) analyses linguistically WordNet glosses (textual description) in order to extract relevant information about a given concept and enrich its properties. This analysis can help detect synonyms and related words and can contribute to concept definition. In fact, concept learning requires not only the identification of conceptual classes but necessitates also concept description through discovery of attributes, sub-classes and relationships. This is further explained in the following sections.

4.2. Attribute Extraction

Since concepts are characterized by a number of features, it is important to discover the distinctive attributes or properties that define a concept. In his ontology, (Guarino, 1992) distinguishes between relational and non-relational attributes. Relational attributes include qualities and relational roles and non-relational attributes include parts. Following (Guarino, 92) and (Pustejovsky, 95), (Almuraheb and Poesio, 2005) presented another scheme for classifying attributes into qualities, parts, related-objects, activities and related-agents.

In this paper, attributes designate a data type property such as *id, name*, etc. in contrast with object properties which are considered as conceptual relationships and addressed in section 4.5.

4.2.1. NLP-based Techniques

According to (Poesio and Almuhareb, 2005), the right meaning of attributes can be found by looking at Wood's linguistic interpretation (Wood, 75): *Y is a value of the attribute A of X if it is possible to say that Y is a A of X (or the A of X).* If it is not possible to find a Y then A cannot be an attribute. In order to comply with this linguistic interpretation, linguistic patterns are also proposed for the detection of attributes. Following (Woods, 1975), (Almuhareb and Poesio, 2005) suggested the use of the following patterns in order to search for attributes of a concept C:

- "(a|an|the) * C (is|was)" (e.g.: *a red car is…*).
- "The * of the C (is|was)" (e.g.: *the color of the car is…*)
- "The C's * R" (e.g.: *The car's price is…*) where R is a restrictor such as "is" and the wildcard denotes an attribute.

(Cimiano, 2006) proposed another set of patterns for attribute extraction based on adjective modifiers and WordNet and presented a number of interesting patterns describing attributes and their range according to syntax (parts-of-speech).

4.2.2. Statistical and machine Learning techniques

As previously said, natural-language processing techniques are generally coupled with statistical filtering and machine learning. (Poesio and Almuraheb, 2005) proposed a supervised classifier for learning attributes based on morphological information, an attribute model, a question model, and an attributive-usage model. These models serve to differentiate different kind of attributes based on a specific classification scheme. In (Poesio and Almuhareb, 2008), the Web is used to extract concept descriptions. Another approach, proposed by (Ravi and Pasca, 2008), describes a weakly supervised classifier for learning attributes and values, based on a small set of examples.

4.3. Taxonomy Extraction

One of the most important tasks in knowledge engineering is the organization of knowledge into taxonomies which indicate generalization/specialization relationships between classes. These relationships enable inheritance between concepts and automated reasoning (Corcho & Gomez-Perez, 2000).

4.3.1. NLP-based Techniques

The most famous way of extracting taxonomical links is the use of specific lexico-syntactic patterns as proposed by Hearst (Hearst, 1992). In Pattern-based approaches, the text is scanned for discovering instances of distinguished lexico-syntactic patterns that indicate a taxonomical link. Patterns are usually expressed as regular expressions (Cimiano and Volker, 2005b) but they can also be represented by dependency relationships (Zouaq and Nkambou, 2009a) (Lin and Pantel, 2001).

Since a domain corpus is sparse and because hierarchical patterns are rare in domain-specific corpora, many approaches extend the corpus by a search of taxonomical links in dedicated resources such as WordNet (Snow et al., 2004) or on the Web (Cimiano et al, 2004) (Maedche and Staab, 2001) so as to increase their recall (Etzioni et al., 2004). In order to remediate the burden of the manual definition of patterns, (Snow et al., 2004) propose a classifier for automatically learning hyponym (is-a) relations from text based on dependency paths and using WordNet.

Other linguistic approaches use the internal structure of multiple-words terms (nouns phrases) in order to deduce taxonomical links. For example, there is a taxonomical link between a term and the same term modified by an adjective (e.g.: an intelligent man is-a man). This approach is quite popular (Buitelaar et al., 2003) (Velardi et al., 2005) (Zouaq and Nkambou, 2009a).

4.3.2. Statistical and machine Learning techniques

Similar to the ones used in concept learning, statistical and machine learning approaches for taxonomy learning rely on Harris' distributional hypothesis. Hierarchical clustering algorithms are used in order to extract taxonomies from text and produce hierarchies of clusters. (Maedche et al., 2002) describe the two main approaches that can be used to implement hierarchical clustering: the bottom-up approach which starts with individual objects and groups the most similar ones, and the top-down approach, where all the objects are divided into groups. This approach has been used by many works such as (Bisson et al, 2000), (Carabello, 99), and (Faure and Nedellec, 98). Typically, as highlighted by (Cimiano et al., 2004), a term t is a subclass of t2 if all the syntactic contexts in which t appears are also shared by t2. The syntactic contexts are used as feature vectors and a similarity measure is applied. For example, in order to compute the relation *is_a (t, t2)*, (Cimiano et al., 2004) applied a directed Jaccard coefficient computing the number of common features divided by the number of features of term t.

(Cimiano et al., 2004) propose also the use of multiple sources of evidence and techniques in order to learn hierarchical relationships. Similarly, (Widdows, 2003) proposes the use of unsupervised methods combining statistical and syntactic information in order to update an existing taxonomy with new terms appropriately.

4.4. *Conceptual Relationships Extraction*

Conceptual relations refer to any relationship between concepts except the taxonomical relations. This include specific conceptual relationships such as synonymy, part-of, possession, attribute-of and causality and more general relationships referring to any labeled link between a source concept (the domain of the relation) and a destination concept (the range of the relation). In the following sections, we identify the different techniques used to describe specific relationships and generic relationships.

4.4.1. NLP-based Techniques

In the information extraction community, conceptual relation extraction is known as template filling, frame filling, semantic role labeling or event extraction. In this case, it relies on lexico-semantic lexicons such as FrameNet (Baker et al., 98) and VerbNet (Kipper et al., 2000) in order to extract particular relationships and to assign roles (such as Agent, Theme, etc.) to the arguments of the relation. Approaches based on frames include ASIUM (Faure and Nédellec, 1998) which enables an acquisition of relations between concepts based on triggering words. Another work related to roles is the identification of Qualia structures by (Pustejovsky, 1995). These qualia structures can help identify particular relationships as showed by (Cimiano and Wenderoth, 2005) who proposed a number of linguistic patterns indicating the different roles defined by Pustejovsky.

There is quite a lot of work related to the use of linguistic patterns to discover particular ontological relations from text. Following Hearst's work (Hearst, 1992) on taxonomic relations, different researchers created patterns for non-hierarchical relationships (Iwanska et al., 2000) (Zouaq and Nkambou, 2009a), for part-of relations (Charniak and Berland, 1999) (Van Hage et al., 2006) or causal relations (Girju et al., 2003). In fact, many works consider that ontological relationships are mostly represented by verbs and their arguments. In the same line of research, (Navigli and Velardi, 2004) use patterns expressed as regular expressions and restricted by syntactic and semantic constraints. Finally, WordNet can be used to extract synonyms, antonyms and other kind of relationships. This involves also the detection of the right sense of the term and thus the use of word sense disambiguation algorithms.

4.4.2. Statistical and machine Learning techniques

Most of the work on relation extraction combines statistical analysis with more or less complex levels of linguistic analysis. For example, (Zouaq and Nkambou, 2009b) exploit typed dependencies in order to learn relationships and statistical measures in order to define if the relationships should be included in the ontology.

There also exist machine learning approaches for learning qualia structures such as the work of (Claveau, 2003) who use inductive logic programming or (Yamada and Baldwin, 2004) who rely on lexico-syntactic patterns but also on a maximum entropy model classifier. (Cimiano and Wenderoth, 2007) developed an algorithm for generating a set of clues for each qualia role, download the snippets of the 10 first Google hits matching the generated clues, part-of-speech-tagging of the downloaded snippets, matching regular expressions conveying the qualia role of interest and finally weighting the returned qualia elements according to some measure.

An interesting approach for learning non labeled relationships is the use of association rule learning, where association rules are created from the co-occurrence of elements in the corpus. This approach has been adopted by the Text-to-Onto system (Maedche and Staab, 2001). However, these relationships should be later manually labeled and this task is not always easy for the ontology engineer.

4.5. Instance Extraction

Instance extraction, also known as Ontology Population (OP), has the objective of finding instances of concepts defined in an ontology, and it is a classification task. It is similar to Named Entity Recognition (e.g. Person, Location, Organization, etc.), which is often used in information extraction. Examples of systems especially devoted to instance extraction include WEB→KB (Craven et al., 2000) and Know-it-All (Etzioni et al., 2004).

4.5.1. NLP-based Techniques

There are a number of approaches that use NLP-based techniques for ontology population. A pattern-based approach similar to the one presented in the taxonomy extraction section relies on Hearst patterns (Hearst, 92) (Schlobach et al., 2004) (Zouaq and Nkambou, 2009b) (Etzioni et al., 2004) or on the structure of words (Velardi et al., 2005). These approaches try to find explicitly stated "is-a" relationships. Other linguistic approaches are based on the definition or the acquisition of rules. For example, the work of (Amardeilh et al., 2005) proposes the definition of acquisition rules that are fired once defined linguistic tags are found. These tags are mapped to concepts, attributes and relationships from the ontology and enable to find instances of these elements.

4.5.2. Statistical and machine Learning techniques

There are supervised and weakly supervised approaches for ontology population (Tanev and Magnini, 2006). Among the weakly supervised approaches, (Cimiano and Volker, 2005b) used vector-feature similarity between each concept c and a term to be categorized t. Cimiano and Volker evaluated different context features (word windows, dependencies) and proved that syntactic features work best. Their algorithm assigned a concept to a given instance by computing the similarity of this instance feature vector and the concept feature vector. (Tanev and Magnini, 2006) used syntactic features extracted from dependency parse trees. Their algorithm required only a list of terms for each class under consideration as training data.

Supervised approaches for ontology population reach higher accuracy. However, they require the manual construction of a training set, which is not scalable (Tanev and Magnini, 2006). An example of a supervised approach is the work of (Fleischman, 2001) (Fleischman and Hovy, 2002) who designed a machine learning algorithm for fine-grained Named Entity categorization. Web->KB (Craven et al., 2000) relies also on a set of training data, which consists of annotated regions of hypertext that represent instances of classes and relations, in order to extract named entities. Based on the ontology and the training data, the system learns to classify arbitrary Web pages and hyperlink paths.

4.6. Axioms Extraction

Axioms extraction represents one of the most difficult tasks of ontology learning. Axioms express necessary and sufficient conditions that are used to constrain the information contained in the ontology and to deduct new information (Shamsfard and Barforoush, 2003). Few systems have tackled the problem of axiom extraction. Among them, HASTI is a system that translates explicit axioms in conditional and quantified natural language sentences to logically formatted axioms in KIF (Shamsfard and Barforoush, 2002). LExO2 (Volker et al, 2008) is another initiative for transforming natural language sentences (definitions) into description logic axioms.

4.6.1. NLP-based Techniques

Natural language techniques for axiom extraction rely on the syntactic transformation of natural language definitions into description logic axioms (Volker et al, 2008). This supposes the availability of such definitions. (Volker et al, 2008) also focus on learning a particular axiom which is disjointness through a lexico-syntactic pattern used to detect enumerations. Their underlying assumption is that terms which are listed separately in an enumeration mostly denote disjoint classes. (Zouaq and Nkambou, 2009b) describe a pattern for defining equivalent classes. This pattern is based on the appositive grammatical relationship between two terms to indicate that these terms are similar and denote the same concept. Another interesting work is the approach of (Lin and Pantel, 2001) which proposes the use of paths in dependency trees in order to learn similar relationships. This enables the creation of inverse properties for these relationships such as *X solves Y* and *Y is solved by X*.

4.6.2. Statistical and machine Learning techniques

To our knowledge, there are very few machine learning approaches for learning axioms. A machine learning classification approach has also been used by (Volker et al., 2008) in order to determine disjointness of any two classes. They extract automatically lexical and logical features providing a basis for learning disjointness by taking into account the structure of the ontology, associated textual resources, and other types of data. The features are then used to build an overall classification model.

5. Ontology Update and Evolution

Despite the important number of initiatives for ontology learning, the results are not still completely satisfactory and the field has to gain more maturity. Moreover, the evolution of ontologies seems to be even less supported in the community. In fact, enabling this evolution (semi) automatically is a key subject matter for the Semantic Web and this involves the ability of updating an ontology with new concepts, relationships, properties and axioms, the ability of appropriately placing a concept in the taxonomy and the ability to perform mapping and alignment between existing ontologies. Here again, we provide a very brief and incomplete glimpse over the NLP-based approaches and the statistical and machine learning approaches. We must also underline that we do not deal with change, versioning and consistency management during the evolution process and we refer the reader to (Haase and Sure, 2004) (Flouris et al., 2006) to gain more insight on this question. Other interesting questions are related to ontology matching and alignment (Shvaiko and Euzenat, 2008) but they are not considered in this paper.

Ontology evolution can target each component of the ontology learning process. From the NLP side, enriching an existing concept with new attributes and relationships has been done in the work of (Velardi et al., 2005) by searching the concept in WordNet and reusing its Synsets in the ontology enrichment. This involves word sense disambiguation. As far as the instance extraction is concerned, we refer the reader to the section 3.6.

From the statistical and machine learning side, there has been attempts to add new concepts in the ontology taxonomy. Updating the ontology with a new concept involves placing it correctly in the hierarchy and retrieving appropriate parents. A number of categorization techniques have been used in order to augment an ontology with a new concept: the *k-nearest neighbor method (kNN),* the *category-based method* and the *centroid-based method* (Maedche et al., 2002). These methods use vector-based features for representing concepts based on co-occurrence and word windows. The new concept can then be placed in the hierarchy according to similarity metrics with existing concepts in the ontology. (Maedche et al., 2002) give a good review about these methods.

6. Current Challenges

There are many challenges that face the ontology engineering community as well as the computer-based educational community that considers the use of ontologies for domain knowledge representation. These can be divided into general challenges and ITS-specific challenges.

Despite the large number of available systems, there is still a need of further developments in ontology engineering. Particularly, there is a need of setting up a reusable framework that enables the combination and comparison of different extraction methods. In fact, there is a lack of reusable services for ontology learning, update and evaluation. There is also a lack of a framework that indicates the available methods for each subtask of the ontology learning process based on various criteria (corpus, task, etc.). Such a framework could enable better informed choices for ontology learning. To my point of view, a service-oriented architecture is essential for a wide development and reuse of automatic methods for ontology learning.

Moreover, one of the problems of automatic methods for ontology learning is that they can produce inconsistent or duplicate entries and dealing with these inconsistencies is a particular challenge (Volker et al., 2008). Inconsistencies can result from the methods used, but also from the input data, which may be too sparse or which may contain contradictions. (Volker et al., 2008) propose three alternatives: using a reasoning-supported process to guarantee that the learned ontologies are kept consistent over time, repairing consistencies after the ontology production or setting up reasoning mechanisms able to deal with these inconsistencies.

Moreover, another challenge is that there is very little support, in the ontology learning tools, regarding many important aspects of ontology engineering especially ontology evolution, reuse, merging, alignment and matching. These different areas still need to mature. Particularly, an important point is to make available a whole environment for ontology engineering involving all the various aspects of the ontology lifecycle.

The general challenges of ontology engineering are all important for the specific ITS-related challenge of building a domain model. In fact, successful attempts for building an ITS domain model automatically have been limited (Suraweera et al., 2004). Ontology engineering can help satisfy this need and contribute to the wide adoption of Intelligent Tutoring Systems. Moreover, ontology engineering can contribute to building a bridge with the eLearning community by making eLearning resources the main material for building the ITS domain model (Zouaq and Nkambou, 2009a). Similarly, eLearning can benefit from this domain model for indexing learning resources and developing more "intelligent" techniques for training learners.

7. Conclusion

We have described the ontology engineering field and particularly ontology learning techniques and we highlighted how intelligent tutoring systems may benefit from this ontology engineering. One of the main advantages of this engineering is that it can provide a solution to two issues: the first one being the difficulty of building an ITS domain model from scratch for each domain and the second one being the difficulty of sharing and reusing the available representations. As standard knowledge representations, ontologies can support the ITS community in producing ITS components more easily and at lower costs. However, this involves the availability of a unified framework for the whole ontology lifecycle including ontology learning, evolution, alignment, matching and evaluation.

8. References

Almuhareb , A. and Poesio, M. (2004). Attribute-based and value-based clustering: an evaluation, *Proc. of EMNLP*, Barcelona, July

Almuhareb, A. and Poesio, M. (2005). Finding Concept Attributes in the Web. *Proc. of the Corpus Linguistics Conference*, Birmingham, July

Amardeilh, F., Laublet, P. and Minel, J-L. (2005). Document annotation and ontology population from linguistic extractions. *Proc. of the 3rd international conference on Knowledge capture*, Banff, Alberta, Canada, pp.161-168.

Aroyo, L., and Dicheva, D. (2004). The New Challenges for E-learning: The Educational Semantic Web. *Educational Technology & Society,* 7(4):59-69.

Aussenac-Gilles, N., Biebow, B. and Szulman, S. (2000). Revisiting Ontology Design: A Methodology Based on Corpus Analysis. *EKAW 2000*, pp. 172-188.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet project. *Proc. of the COLING-ACL,* Montreal, Canada.

Bisson, G., Nedellec, C. and Canamero, L. (2000). Designing clustering methods for ontology building – The Mo'K workbench. *Proc. of the ECAI Ontology Learning Workshop*, pp. 13-19.

Buitelaar, P., Olejnik, D. and Sintek, M. (2003). A protege plug-in for ontology extraction from text based on linguistic analysis. *Proc. of the International Semantic Web Conference (ISWC).*

Buitelaar, P., Cimiano, P., and Magnini, B. (2005). Ontology Learning from Text: An Overview In: Paul Buitelaar, Philipp Cimiano, Bernardo Magnini (eds.) *Ontology Learning from Text: Methods, Evaluation and Applications*, Frontiers in Artificial Intelligence and Applications Series, Vol. 123, IOS Press, July 2005.

Caraballo, S.A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. *Proc. of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 120-126.

Charniak, E. and Berland, M. (1999). Finding parts in very large corpora. *Proc. of the 37th Annual Meeting of the ACL*, pp. 57–64.

Cimiano, P. (2006). Ontology Learning Attributes and Relations, In Ontology Learning and Population from Text, pp. 185-231, Springer.

Cimiano, P., and Wenderoth, J. (2007). Automatic Acquisition of Ranked Qualia Structures from the Web. *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics (ACL).* Prague.

Cimiano, P., and Wenderoth, J. (2005). Automatically Learning Qualia Structures from the Web. *Proc. of the ACL Workshop on Deep Lexical Acquisition*, pp. 28-37, 2005.

Cimiano, P. and Völker, J. (2005a). Text2Onto–A Framework for Ontology Learning and Data-driven Change Discovery. *Proc. of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB),* pp. 227-238. Alicante:Springer.

Cimiano, P. and Volker, J. (2005b). Towards large-scale, open-domain and ontology-based named entity classification. *Proc. of RANLP'05,* pp. 166–172, Borovets, Bulgaria.

Cimiano, P., Pivk, A., Schmidt-Thieme, L., and Staab, S. (2004). Learning Taxonomic Relations from Heterogeneous Sources of Evidence. In *Ontology Learning from Text: Methods, Evaluation and Applications.* IOS Press.

Cimiano, P. and Staab, S. (2004). Learning by googling, *ACM SIGKDD Explorations*, 6(2):24-33.

Claveau, V. (2003). Acquisition automatique de lexiques sémantiques pour la recherche d'information. Thèse de doctorat, Université de Rennes-1, Rennes.

Corcho, O, and Gómez-Pérez, A. (2000). A Roadmap to Ontology Specification Languages. *Proc. of EKAW 2000*, pp. 80-96.

Craven, M, DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T.M., Nigam, K. and Slattery, S. (2000). Learning to construct knowledge bases from the World Wide Web, *Artificial Intelligence*, 1-2 (118): 69-113.

Deiltel, A. , Faron-Zucker, C. and Dieng, R. (2001). Learning ontologies from rdf annotations, Proc. of the IJCAI Workshop in Ontology Learning.

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D.S. and Yates, A. (2004). Web-scale information extraction in KnowItAll (preliminary results). *Proc. of the 13th World Wide Web Conference*, pp. 100–109.

Faure, D. and Nedellec, C. (1998). A corpus-based conceptual clustering method for verb frames and ontology. In P. Velardi, editor*, Proc. of the LREC Workshop on Adapting lexical and corpus resources to sublanguages and applications*, pp. 5–12.

Frantzi, K., Ananiadou, S. and Tsuji, J. (1998). The c-value/nc-value method of automatic recognition for multi -word terms. *Proc. of the ECDL,* pp 585–604.

Fleischman, M. (2001). Automated Subcategorization of Named Entities. In 39th Annual Meeting of the ACL, Student ResearchWorkshop, Toulouse, France, July.

Fleischman, M. and Hovy, E. H. (2002). Fine Grained Classification of Named Entities. *COLING 2002*

Flouris, G., Plexousakis, D. and Antoniou, G. (2006). Evolving Ontology Evolution. In *Proc. of the 32nd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM-06),* Invited Talk, pp. 14-29.

Fortuna, B. , Grobelnik, M. and Mladenic, D. (2007). OntoGen: Semi-automatic Ontology Editor. *HCI International 2007*, Beijing.

Fortuna, B., Mladevic, D.and Grobelnik, M. (2005). Visualization of Text Document Corpus. In *ACAI 2005 Summer School*, 2005.

Garshol, L. M. (2004). Metadata? Thesauri? Taxonomies? Topic Maps! Making Sense of it all, In *Journal of Information Science,* 30(4): 378-391.

Girju, R., Badulescu, A., and Moldovan, D. (2003). Learning Semantic Constraints for the Automatic Discovery of Part –Whole Relations. *Proc. of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language*, pp. 1-8, Morristown: Association for Computational Linguistics.

Guarino, N. (1992). Concepts, attributes and arbitrary relations: some linguistic and ontological criteria for structuring knowledge base. *Data and Knowledge Engineering*, 8, pp. 249–261.

Guarino, N. and Welty, C. (2002). Evaluating Ontological Decisions with OntoClean. *Communications of the ACM,* 45(2):61-65, New York:ACM Press

Guin-Duclosson, N., Jean-Danbias, S., and Norgy, S. (2002). The AMBRE ILE: How to Use Case-Based Reasoning to Teach Methods. In Cerri, S.A., Gouarderes, G., Paraguacu, F. (eds.): *Sixth International Conference on Intelligent Tutoring Systems.* Lecture Notes in Computer Science, Vol. 2363, pp.782-791, Springer-Verlag, Berlin.

Haase, P., Sure, Y. (2004). State-of-the-Art on Ontology Evolution, SEKT deliverable 3.1.1.b, available at http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/SEKT-D3.1.1.b.pdf

Harris, Z. (1954). Distributional structure. *Word*, 10(23): 146-162.

Hatzilygeroudis, I and Prentzas, J. (2004). Knowledge Representation Requirements for Intelligent Tutoring Systems. *Intelligent Tutoring Systems* pp. 87-97

Hearst, M. (1992). Automatic Acquisition of Hyponyms from LargeText Corpora. *Proc. of the Fourteenth International Conference on Computational Linguistics*,pp.539-545, Nantes.

Hearst, M. and Schutze, H..(1993). Customizing a lexicon to better suit a computational task. *ACL SIGLEX Workshop,* Columbus, Ohio.

Hindle, D. (1990). Noun classification from predicate-argument structures. *Proc. of ACL-90*, pp. 268–275, Pittsburg, Pennsylvania.

Hsieh, P., Halff, H., and Redfield, C. (1999). Four easy pieces: Developing systems for knowledge-based generative instruction. *Int. Journal. of Artificial Intelligence in Education. 10: 1-45.*

Iwanska, L.M., Mata, N., and Kruger, K.(2000). Fully automatic acquisition of taxonomic knowledge from large corpora of texts. *Natural Language Processing and Knowledge Processing*, pp.335–345. MIT/AAAI Press.

Jannink, J. and Wiederhold, G. (1999). Ontology maintenance with an algebraic methodology: A case study. In *Proc. of AAAI workshop on Ontology Management*.

Kabbaj, A., Rouane, K. and Frasson, C. (1996). The use of a semantic network activation language in an ITS project, *Proc. of the Third International Conference on Intelligent Tutoring Systems*, pp. 279 – 287.

Kipper, K. Dang, H.D., Palmer, M. (2000). Class-Based Construction of a Verb Lexicon. *Proc. of AAAI-2000 Seventeenth National Conference on Artificial Intelligence,* pp. 691 - 696.

Kumar, A. (2006). Using Enhanced Concept Map for Student Modeling in a Model-Based Programming Tutor, in *Proc. of 19th International FLAIRS Conference on Artificial Intelligence,* Melbourne Beach, FL.

Lin, D. and Pantel, P. (2001). Induction of semantic classes from natural language text. *Proc. of SIGKDD-01,* pp. 317-322. San Francisco, CA.

Lin, D. (1999). Automatic identification of non-compositional phrases. *Proc. of ACL 1999*, pp. 317-324.

Lin, D. (1998). Automatic Retrieval and Clustering of Similar Words. In *Proc. of COLING-ACL 98.* pp. 768-774. Montreal, Canada.

Maedche, A., Pekar, V. and Staab, S. (2002). Ontology Learning Part One - On Discovering Taxonomic Relations from the Web. *Web Intelligence*, pp. 301-322, Springer.

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

Moldovan, D.I. and Girju, R.C. (2001). An interactive tool for the rapid development of knowledge bases. *International Journal on Artificial Intelligence Tools (IJAIT)*, 10(1-2).

Navigli, R., Velardi, P., Cucchiarelli, A., and Neri, F. (2004). Quantitative And Qualitative Evaluation Of The OntoLearn Ontology Learning System, *Proc. of the 20th international conference on Computational Linguistics,* Switzerland.

Navigli, R., and Velardi, P. (2004). Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. *Computational Linguistics*, 30(2): 151-179.

Nkambou, R. (1999). Managing Inference Process in Student Modeling for Intelligent Tutoring-Systems. *Proc. of the Eleventh IEEE International Conference on Tools with Artificial intelligence*. IEEE Computer Society Press.

Nyulas, C., O'Connor, M., Tu, S. (2007). Datamaster - a plug-in for importing schemas and data from relational databases into protégé. In Proc. of 10th Intl. Protégé Conference, Budapest.

Papatheodorou, C., Vassiliou, A., Simon, B. (2002): Discovery of Ontologies for Learning Resources Using Word-based Clustering. *Proc. of ED-MEDIA 2002*. AACE, Denver, USA.

Poesio, M., and Almuhareb, A. (2005). Identifying Concept Attributes Using A Classifier. *Proc. of the ACL Workshop on Deep Lexical Acquisition*, pp. 18-27, Ann Arbor: Association for Computational Linguistics.

Poesio, M. and Almuhareb, A. (2008) Extracting concept descriptions from the Web: The importance of attributes and values. In Buitelaar P. and P. Cimiano (eds.) *Bridging the Gap between Text and Knowledge,* pp. 29-44, IOS Press.

Polson, M. C. and Richardson, J.J., Eds. (1988). Foundations of Intelligent Tutoring Systems. L. Erlbaum Associates Inc.

Pustejovsky, J. (1995). The generative lexicon. MIT Press.

Ravi, S. and Pasca, M. (2008). Using Structured Text for Large-Scale Attribute Extraction, *Proc. of the 17th ACM Conference on Information and Knowledge Management (CIKM-2008).*

Reinberger M.-L and Spyns P. (2005). Unsupervised Text Mining for the learning of DOGMA-inspired Ontologies, In, Buitelaar P., Cimiano Ph., & Magnini B.,(eds.), *Ontology Learning from Text: Methods, Applications and Evaluation., Advances in Artificial Intelligence*, pp. 29 - 43. IOS Press, Amsterdam.

Rigau G., Rodríguez H. and Agirre E (1998). Building Accurate Semantic Taxonomies from Monolingual MRDs. *Proc. of the 17th International Conference on Computational Linguistics* COLING-ACL'98. Montreal, Canada.

Sabou, M. (2005) Learning Web Service Ontologies: an Automatic Extraction Method and its Evaluation, *Ontology Learning from Text: Methods, Evaluation and Applications*, eds. P.Buitelaar, P. Cimiano, B. Magnini, IOS Press.

Sahlgren, M. (2006). The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. Ph.D. dissertation, Department of Linguistics, Stockholm University.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):515-523.

Schlobach, S., Olsthoorn, M. and de Rijke, M. (2004). Type Checking in Open-Domain Question Answering. *Proc. of European Conference on Artificial Intelligence,* 398-402, IOS Press.

Shamsfard, M., & Barforoush,A.A.(2003).The State of the Art in Ontology Learning: A Framework for Comparison. *The Knowledge Engineering Review,* 18(4): 293-316.

Shamsfard M., and Barforoush, A. A., (2002) An Introduction to HASTI: An Ontology Learning System, *Proc. of 6th Conference on Artificial Intelligence and Soft Computing (ASC'2002)*, Banff, Canada.

Shvaiko, P. and Euzenat, J. (2008). Ten Challenges for Ontology Matching. *OTM Conferences*, pp. 1164-1182.

Snow, R, Jurafsky, D. and Ng, A. Y. (2004) Learning syntactic patterns for automatic hypernym discovery.*Advances in Neural Information Processing Systems (NIPS 2004)*, Vancouver, British Columbia.

Stojanovic, L., Stojanovic, N., and Volz R. (2002). Migrating data-intensive Web Sites into the Semantic Web. *Proc. of the 17th ACM symposium on applied computing (SAC)*, ACM Press, 2002, pp. 1100-1107.

Suraweera, P., Mitrovic, A, Martin, B. (2004) The The role of domain ontology in knowledge acquisition for ITSs. Maceió, Alagoas, Brazil: Intelligent Tutoring Systems: *7th International Conference (ITS 2004)*. Lecture Notes in Computer Science, 3220, pp. 207-216.

Suryanto H. and Compton, P. (2001). Discovery of Ontologies from Knowledge Bases. *Proc. of the First International Conference on Knowledge Capture, Eds.* Yolanda Gil; Mark Musen; Jude Shavlik, Victoria, British Columbia Canada, pp. 171-178, The Association for Computing Machinery, New York, USA.

Tanev, H. and Magnini, B. (2006), Weakly Supervised Approaches for Ontology Population. *Proc. of EACL'06*, pp. 3-7, Trento, Italy.

Van Assem, M., Menken, M. R., Schreiber, G., Wielemaker, J. and Wielinga, B. J. (2004) A Method for Converting Thesauri to RDF/OWL. *International Semantic Web Conference* pp. 17-31.

Van Hage, W. R., Kolb, H., & Schreiber, G. (2006). A Method for Learning- Part-Whole Relations. *Proc. of the 5th International Semantic Web Conference*, pp. 723-735, Berlin, Heidelberg: Springer.

Van lehn, K., Zhendong, N. (2001). Bayesian student modeling, user interfaces and feedback: a sensitivity analysis. *International Journal of AI in Education* 12: 155-184.

Vassileva, J. (1998). Dynamic Course Generation on the WWW. *British Journal of Educational-Technologies* 29: 5-14.

Velardi, P., Navigli, R., Cuchiarelli, A.and Neri, F. (2005). Evaluation of ontolearn, a methodology for automatic population of domain ontologies. In P. Buitelaar, P. Cimiano, and B. Magnini, editors, *Ontology Learning from Text: Methods, Applications and Evaluation*. IOS Press.

Volker, J., Haase, P. and Hitzler, P. (2008). Learning Expressive Ontologies. In Paul Buitelaar and Philipp Cimiano, *Ontology Learning and Population: Bridging the Gap between Text and Knowledge,* volume 167 of Frontiers in Artificial Intelligence and Applications, pp. 45-69. IOS Press, Amsterdam.

Volz, R., Oberle, D., Staab, S., and Studer, R. (2003). OntoLiFT Prototype. IST Project 2001-33052 WonderWeb Deliverable 11.

Widdows D. (2003).Unsupervised methods for developing taxonomies by combining syntactic and statistical information. *Proc. of HLT-NAACL* pp. 197-204.

Widdows, D. and Dorow, B. (2002). A graph model for unsupervised lexical acquisition. In *19th International Conference on Computational Linguistics*, pages 1093–1099, Taipei, Taiwan.

Woods, W. A. (1975). What's in a link: Foundations for semantic networks. In Daniel G. Bobrow and Alan M. Collins, editors, *Representation and Understanding: Studies in Cognitive Science*, pp. 35-82. Academic Press, New York.

Yamada, I., and Baldwin, T. (2004). Automatic discovery of telic and agentive roles from corpus data. *Proc. of the 18th Pacific Asia Conference on Language, Information and Computation (PACLIC),* pp.115–126.Tokyo.

Zouaq, A. and Nkambou, R. (2009a), "Enhancing Learning Objects with an Ontology-Based Memory," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 6, pp. 881-893.

Zouaq, A. and Nkambou, R. (2009b), "Evaluating the Generation of Domain Ontologies in the Knowledge Puzzle Project," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 11, pp. 1559-1572.