Document Semantic Annotation for Intelligent Tutoring Systems: a Concept Mapping Approach

Amal Zouaq[1], Roger Nkambou[2], Claude Frasson[1]

[1]University of Montreal, [2]UQAM
[1]CP 6128, Succ. Centre-Ville, Montreal, QC, H3C3J7
[2]CP 8888, Succ. Centre-Ville, Montreal, QC, H3C3P8
zouaq@iro.umontreal.ca, nkambou.roger@uqam.ca, frasson@iro.umontreal.ca

Abstract

The difficulty of domain knowledge acquisition is one of the most sensible challenges of intelligent tutoring systems. Relying on domain experts and building domain models from scratch are not viable solutions. The ability to automatically extract domain knowledge from documents can contribute to overcome these difficulties. In this paper, we use machine learning and natural language processing to parse documents and to generate domain concept maps and ontologies. We also show how an intelligent tutoring system benefits from the generated structures.

1. Introduction

Documents have always been a privileged vehicle of knowledge within human communities. In fact, documents within communities of practice constitute a rich unstructured knowledge base often neglected when trying to mine knowledge. Thus the creation of semantic metadata related to document content seems to be the only way to exploit this knowledge base and extract implicit and explicit expertise.

Semantic annotation of documents hence constitutes one of our pursued goals. Semantic annotation refers to the process of indexing and retrieving useful knowledge from documents thus creating metadata. The semantic web can help us attain this goal as it proposes the use of domain ontologies to annotate document content (Berners-Lee, Hendler and Lassila 2001). However, the development of a domain ontology is not a trivial task and consumes important resources in term of time and money. Thus, (semi) automatic generation of domain ontologies should be realized to reduce the cost of such an operation.

An intelligent tutoring system (ITS) could greatly benefit from such a domain ontology. In fact, knowledge acquisition has always been the major bottleneck for intelligent tutoring systems (Murray 2003). Therefore, easy tools must be elaborated for knowledge acquisition and authoring to make intelligent tutoring systems more widespread in academic and industrial settings.

Our objective is to handle all the process of knowledge acquisition and dissemination: beginning from documents, we perform annotations on these documents as well as knowledge extraction and ontology generation. Then we show how the generated concept maps can be reused by an intelligent tutoring system to provide efficient training.

The paper is organized as follows: First, we present the state of the art in the domain of concept maps, intelligent tutoring systems, and ontology generation from text. Then we talk about our architecture and our approach for competence edition, concept map extraction and domain ontology exploitation. Finally, we finish with a conclusion as well as our intended future work.

2. State of the Art

2.1. Concept Mapping and Intelligent Tutoring Systems

Since their inception, concept maps have proven to be very useful representations for knowledge acquisition and elicitation. They provide a synthesized view of domain knowledge and serve as individual and collective knowledge elicitation and capture tools, hence transforming tacit knowledge into explicit one. This enables the knowledge retention within the community. They facilitate the sharing of a common model within a community (which is the main goal of ontologies) and the communication and summarization of complex schemas. Finally, they contribute to meaningful learning (Wikipedia 2006). Indeed, the interest of concept maps in training environments is not to be proven (Novak and Cañas 2006). Intelligent tutoring systems can benefit from domain concept maps for their domain model and their learner model (Kumar 2006). For example, in an intelligent tutoring system, a domain agent can exploit concepts and relationships to provide useful explanations. In previous work (Zouaq, Frasson, and Rouane 2000), we already used concept mapping and visual representations to structure domain knowledge and more precisely explanations and to identify both valid and invalid ideas held by students. The use of appropriate instructional strategies, which exploit concept's linked structure, helps a learner relate his previous knowledge to a new one.

Many concept maps editing tools have been developed such as the IHMC CmapTools (Cañas *et al.* 2004) or the VUE tool (Kumar and Kahle 2006). These tools enable the construction of concept maps and their annotation with multimedia material (notes, audio, images, etc.). However, besides the edition of concept maps, we believe that a semi-automatic generation process must be supported. The Semantic Web and ontology generation from text contribute to this goal.

2.2. Ontology Generation from Text

Due to the recent advances in the natural language processing and machine learning fields, ontology generation from text is becoming a very important area of ontology engineering. Indeed, manual generation of ontologies is a very time-intensive and error-prone process. Moreover, it is difficult to maintain such ontologies and make them reflect recent changes and advances of their domain. For this reason, researchers try to find semi-automatic and automatic ways to generate domain ontologies. This task is not an easy one. Automatic ontology generation requires a deep language understanding and a recognition of the language ambiguities, which is not yet very well realized by natural language processing technologies.

In (Buitelaar, Cimiano, and Magnini 2005), the authors present an interesting overview of the ontology generation layers. According to them, it consists of six extraction layers of growing complexity: terms, synonyms, concepts, taxonomy, relations and rules. A number of systems have been proposed for ontology learning from text. These systems combine one or more of the six layers mentioned above. Examples of systems are InfoSleuth (Hwang 1999), Text-To-Onto (Maedche and Staab 2000), Ontolearn (Navigli, Velardi, and Gangemi 2003), OntoLT (Buitelaar, Olejnik, and Sintek 2004) and GlossOnt (Park 2004). Most of these systems exploit linguistic analysis and machine learning algorithms to find interesting **concepts** and **relationships**.

For example, in **Text-To-Onto** (Maedche and Staab 2000), a collection of domain documents is annotated linguistically with NLP tools and a number of occurring terms are extracted. Then, using an association rules algorithm, which finds correlations in the co-occurrence of classes of terms, the system identifies possible relations between these terms. Finally, the system represents these terms and relations as classes in the ontology.

In **InfoSleuth** (Hwang 1999), human experts provide a small number of high-level concepts that are used to automatically collect relevant documents from the web. Then the system extracts phrases containing these concepts, generates corresponding concept terms and stores them in the ontology. InfoSleuth extracts several kinds of relations such "is-a" and the "assoc-with" relation, which is used to define all the relationships that are not explicitly modeled. For each iteration, a human expert is consulted to verify the correctness of the concepts.

In **GlossOnt** (Park 2004), the author proposes a semi-automatic method for building *partial* ontologies, which focuses on a particular domain concept at a time and which represents only domain concepts and relationships regarding the target concept. The proposed method takes a target concept from the user, searches knowledge sources about the target concept, such as domain glossaries and web documents, and extracts ontological concepts and relationships that are relevant to the target concept.

More recent systems deal with the uncertain and possibly inconsistent knowledge of the automatically generated ontologies. In (Haase and Volker 2005), the authors present **Text2Onto**, a framework to generate OWL ontologies from learned ontology models (LOM) by considering the uncertainty of the knowledge as annotations. Annotations represent the confidence about the correctness of the ontology elements. Then they transform the LOM model to a standard logic-based ontology language in order to be able to use standard reasoning over the learned ontologies.

3. Our Layered Model

We developed a layered model that uses natural language processing to extract concept maps from documents and that organizes the generated knowledge into OWL ontologies. We consider that concept maps represent a sort of *lightweight domain ontology*. They carry out more meaning than simple taxonomies because the extracted relationships between concepts are not only hierarchical ones. They do not try to model a whole domain (but only the text from which they are extracted) and they lack concept's attributes but in the same time a more complete domain ontology (with classes and attributes) can also be considered as a concept map. Therefore, this paper presents the production of this kind of lightweight ontology, but we are already engaged in the process of building a more complete domain ontology through natural language processing.

Our approach is inspired from the concept maps and topic maps fields. In its philosophy, it is very near from the topic map approach as the extracted concepts can be more considered as "topics" with occurrences and associations. Moreover, topic maps are used for indexing and information retrieval purposes, and so are our document concept maps used for. However, our approach differs from topic maps in the sense that we do not comply to the Topic Map ISO standard or to the TMAPI API (TMAPI 2006). In fact, we use the web ontology language (OWL) to model our document concept maps. The similarity to the concept map approach lies in the fact that we use a semantic network representation to model a domain knowledge. Concept maps are used in training and learning which is our aim. However, concept maps use also a top-down approach in constructing the actual map from a more general concept to a more specialized one. Even if it is desirable, this is not mandatory in our system. In fact, the automatic approach generates a semantic network of concepts and links and a human expert can redesign the network in order to generalize some concepts or specialize others. The resulting document concept map shows the natural progression of the textual content.

Figure 1 depicts our architecture.

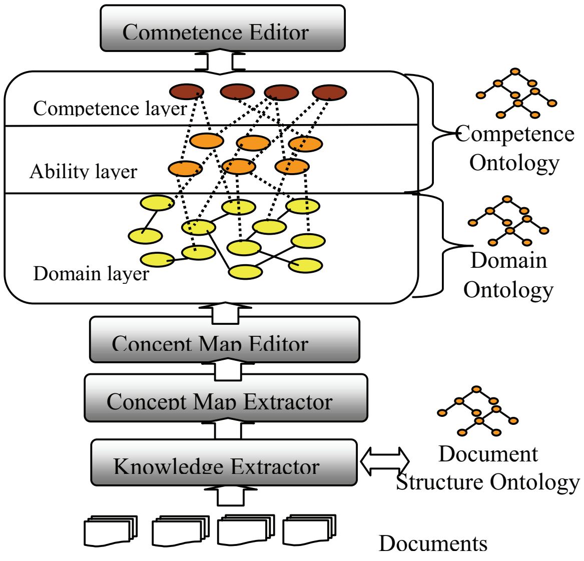


Figure 1: Our Architecture

The interconnected layers enable to define a competence according to a set of concepts and to retrieve document's portions indexed by their concepts and relationship. The domain ontology represents the concept maps obtained through an extraction and an edition process. The competence ontology is used to indicate the training needs in term of abilities.

3.1. Competence Edition

Due to the final aim of our system, which is training, a competence is represented as a learning objective. A learning objective is described as the set of abilities to be mastered by a learner after a pedagogical activity (Nkambou, Frasson, and Gauthier 2003). The abilities are classified, in our model, according to the Bloom's taxonomy (Bloom 1956). Bloom defined six levels of intellectual behavior important in learning and associated a set of action verbs to each level. Thus an ability, which represents a level, is qualified by these action verbs. Bloom's taxonomy enables the definition of competencies at a very detailed level. Our competencies are stored in the Competence Ontology and are created through a competence editor.

A competence is a set of abilities defined on domain concepts. For example, in a learning objective about the SCORM Content Aggregation Model (CAM), the competence can be composed of two abilities: "**define** SCORM CAM" and "**describe** SCORM components". The abilities in the example are indicated in bold and correspond to the levels of acquisition and comprehension. *SCORM CAM* and *SCORM Components* are concepts of the domain ontology.

3.2. Document Concept Map Extraction

We need to be able to reuse document content for information retrieval and training purposes. Thus we need to build a **concept map** for each document (or a partial ontology as stated by Park (Park 2004)) showing the important concepts and relations between them.

Our global approach is the following: First, the system identifies document's paragraphs and sentences using IBM's unstructured Information management architecture (UIMA 2006). Once the document structure is obtained, it uses a machine learning algorithm to find document's keywords. Then the system builds a conceptual network for these keywords by collecting the sentences containing them and parsing them through a statistical NLP parser. A set of transformation rules is applied to the grammatical categories obtained through the parsing process, and triples of the form concept-verb-concept are extracted as well as other types of relations expressing time, place, etc. Indeed, verbs express central semantic relations between concepts and specify the interaction between their subjects and objects. We agree with the fact that domain ontologies rarely model verbs as relations between concepts (Schutz and Buitelaar 2005). When they do so, the verbs are already modeled as a property with a domain and range in the ontology, and the mining process tends to discover occurrences of these verbs. In our case, we do not have a set of predefined verbs or relations. Our aim is to discover all the possible verbal relations between concepts. These relations can serve us in the training process to deploy multiple pedagogical strategies. For example, they can serve to give a conceptual overview of a subject area, or to make connections between two learned concepts thus enlightening a tacit link, etc.

The whole process builds a document concept map that enables to index document content. The union of all the document concept maps constitutes our domain ontology. As Park stated (Park, 2004), we think that this approach is more feasible than methods that try to build a *full* ontology from a collection of documents. In fact, the system intends to focus on a small number of domain concepts which are the document keywords and identifying target concepts and relationships in documents can thus be more focused. This approach can produce more up-to-date ontologies because a document collection within a community is rapidly evolving and new documents can easily be processed.

3.2.1. Document Structure Extraction

Based on IBM's Unstructured Information Management Architecture (UIMA 2006), we developed annotators for detecting sentences and paragraphs. UIMA is an integrated solution that analyzes unstructured information to discover, organize, and deliver relevant knowledge to the end-user.

For the moment, only paragraphs and sentences are determined automatically, but a human expert can manually annotate other structural types such as sections, images, tables, etc. The expert must also indicate the instructional role of the sentences and/or the paragraphs in term of abilities on domain concepts (like the competences). For example, a paragraph can ***describe*** a domain concept, or ***define*** it.

The resulting structures are stored in a **Document Structure Ontology**.

3.2.2. Keyword Extraction

Like the project InfoSleuth (Hwang 1999), we rely on seed words to begin the mining process. Unlike InfoSleuth where a human expert provides these keywords, we use a machine learning algorithm named Kea-3.0 (Frank et al. 1999) to discover document's keywords.

Kea-3.0 is a key phrase (one or more words) extraction algorithm developed by members of the New Zealand Digital Library Project. Its is composed of two phases. The first one is the training phase, where Kea acquires a Naïve Bayesian model from a set of training documents with their author-supplied key phrases. The extraction phase begins once a training model is available to extract key phrases from new documents. Each document is converted to text format and all its candidate phrases are extracted and converted to their canonical form. Many are immediately discarded. The distance of the phrase's first appearance in the document and the TF*IDF features are computed for the remaining phrases. The model uses these attributes to calculate the probability that each candidate phrase is a key phrase. The most probable candidates are output in ranked order and constitute the document key phrases.

When keywords are determined, sentences containing them are collected and analyzed through natural language processing.

3.2.3. Natural Language Processing

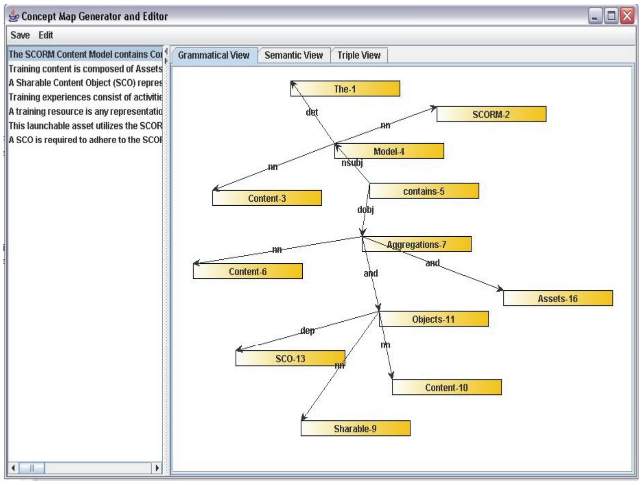
A number of natural language processing tools have been developed among them are the probabilistic parsers which rely on hand-parsed sentences to be trained and which try to produce the most probable analysis of new sentences (De Marneffe, MacCartney, and Manning 2006).

We used the Stanford Parser (Klein and Manning 2003) that is among the Treebank-trained statistical parsers and is capable of generating parses with high accuracy. More specifically, we used its typed dependency parsing component (De Marneffe, MacCartney, and Manning 2006) to parse our candidate phrases (sentences containing keywords extracted by Kea-3.0). According to (De Marneffe, MacCartney, and Manning 2006), typed dependencies and phrase structures represent the structure of sentences in a different way: a phrase structure parse represents nesting of multi-word constituents whereas a dependency parse represents dependencies between individual words labeled with grammatical relations, such as subject, direct object or noun compound modifier.

We developed a Graphical Concept Map Editor and Generator that enables to view the results of the typed dependency parses described above (Figure 2). In the left, you can see the set of candidate key phrases, and the typed dependency structure is in the right pane.

From this typed dependency structure, we created a set of transformation rules or patterns:

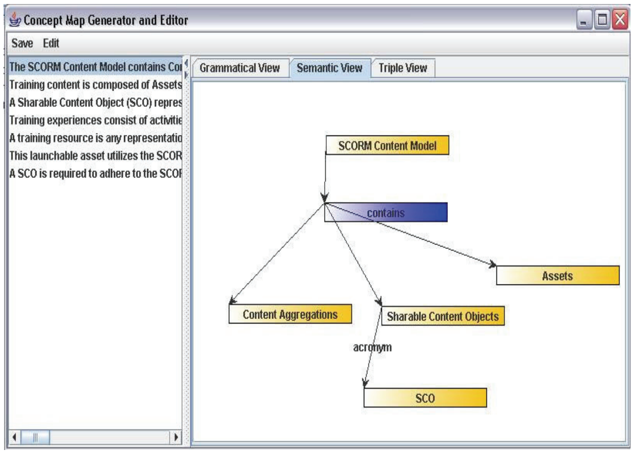
- To aggregate two or more related words into a single concept. For example, if a word has two noun compound modifiers, than the three nodes correspond to one concept (for example: SCORM Content Model, SCORM and Content being noun compound modifiers for the word Model);
- To delete some words such as determiners or "that" and "which" nodes;
- To search acronyms from dependent relations ("dep");
- To conserve relations that indicate prepositions and conjunctions;
- To identify verbs and their auxiliary (for active and passive forms);
- To convert a node verb into a semantic verbal relation;
- To contract some nodes and their grammatical relation for example: a node verb "is inserted" and an "into" relation thus creating a single verbal relation (is inserted into).



The SCORM Content Model contains Co
Training content is composed of Assets
A Sharable Content Object (SCO) repres
Training experiences consist of activitie
A training resource is any representatio
This launchable asset utilizes the SCOR
A SCO is required to adhere to the SCOF

Figure 2: A Grammatical Concept Map

The application of these transformation rules offers a semantic view of the previous structure as indicated in the figure 3.



The SCORM Content Model contains Co
Training content is composed of Assets
A Sharable Content Object (SCO) repres
Training experiences consist of activitie
A training resource is any representatio
This launchable asset utilizes the SCOR
A SCO is required to adhere to the SCOF

Figure 3: A Semantic Concept Map

For example, for the sentence "The SCORM content model contains Content Aggregations, Sharable Content Objects (SCO) and Assets", we obtain the concepts:

- SCORM Content Model
- Content Aggregations
- Sharable Content Objects
- SCO
- Assets

We also obtain two semantic relations:

- The "Acronym" relation between Sharable Content Objects and SCO: **Acronym** (Domain: Concept, Range: Concept)
- The verbal relation "contains" between SCORM Content Model and the three concepts: Assets, Sharable Content Objects and Content Aggregations.

As we already said, Domain Ontology represents the union of all the document concept maps obtained through the automatic ontology generation. Figure 4 shows a partial view of the generated domain ontology.

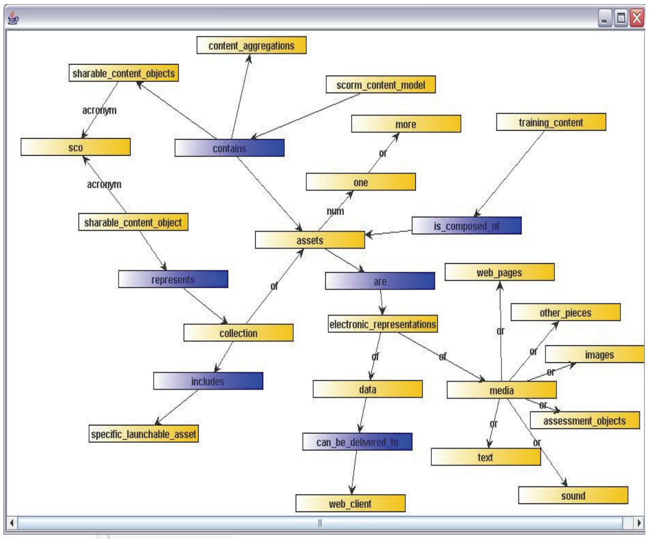


Figure 4: A Domain Concept Map around the Concept of Assets

3.3. Document Concept Map Edition

We also provide editing tools to modify the concept maps and to create courses from the generated domain ontologies.

A **Concept Map Editor** shows a document and its concept map. A human expert is then able to ascertain the correctness of the concept map, to modify it to make it more conformant to his desires or to correct mistakes coming from the statistical parsing. The expert is able to add a new concept, a new relation, to contract a concept and its relation, to divide a concept in two or more concepts, etc. He also has layout facilities to explore the map. For example, the selection of a concept or a relationship highlights them in the document. The selection of a concept related to a sentence highlights the whole sentence in the concept map and in the document. The expert is also provided with tools to explore the knowledge base. For instance, he can search for other documents related to the current selected concept.

4. Domain Ontology Exploitation

The exploitation of such generated domain ontology by an intelligent tutoring system takes different forms.

From a course authoring point of view, a **Course Editor** enables a human designer to create courses from the generated domain ontology in the form of course concept maps. Starting from a competence definition and its abilities, the system searches all the structures related to the required abilities on domain concepts. Following our last example, for a competence that must explain the notion of assets, the system retrieves the semantic map in figure 4. Besides the actual definition and explanation of assets, he has a strong support about what notions must be incorporated in this context, for example, the notions of SCORM Content Model, of training content and of sharable content objects. Then he can save the whole retrieved concept map or part of it as a course concept map. He can explore the different documents related to the generated map and he can also attach to each concept a new pedagogical resource (audio, image, text, video). He can also delete part of the map, or add new concepts.

From a learning point of view, the learner is provided with the concept map that best suits his needs. A competence gap analysis is performed to determine what he already knows in the concept map, as well as the required prerequisites. Two presentation modes can be used:

- ***Concept map oriented Mode:*** The learner is able to access the documents from which the concept under study was extracted or found through ontological indexing. Documents represent the context of their concept map.
- ***Learning Object Oriented Mode:*** a learning object is automatically assembled to fit the competence definition and the learner needs. If multiple content objects (documents, paragraphs, sentences, etc) can be used to master the same ability, the system chooses the content object that has achieved the best performance results for other previous learners. If such performance data does not exist, the system chooses a content object randomly. The others are kept as remediation resources in case of bad results in exercises or in case of an explicit learner demand for additional resources about the ability. Moreover, the available concept maps serve us in the dynamic generation of exercises. We ask the learner to build his own concept map from the list of available concepts and links. As we already said, this can help us identify student's misunderstandings.

5. Conclusion and Further Work

Concept maps are a valuable resource whose rich structure can be exploited in information retrieval and in training, and particularly by intelligent tutoring systems. We presented a solution to automatically generate concept maps from domain documents hence constituting de facto a domain ontology. We also showed how an intelligent tutoring system can benefit from the generated structures to provide efficient training. Indeed, the interest of concept maps in training have already been measured and proven (Novak and Cañas 2006). The interest of our approach is that the generated structures are not coupled to a particular domain knowledge.

More work must be done to mine concept's attributes through natural language processing and to enhance the semantics of the domain ontology through a linguistic knowledge base. Efforts must also be devoted to a complete evaluation of the domain ontology. Moreover, we are working on the generation of more complex learning objects that exploit the concept maps as well as instructional theories to fulfill a learning objective.

6. References

Berners-Lee, T., Hendler, J., and Lassila, O. 2001. The Semantic Web, *Scientific American,* pp.34–43.

Bloom, B. S. 1956. *Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain*. New York: Longman.

Buitelaar, P., Cimiano, P. and Magnini, B. 2005. Ontology Learning from Text: An Overview. In P. Buitelaar, P. Cimiano, B. Magnini (Eds.), *Ontology Learning from Text: Methods, Evaluation and Applications*, volume 123 of Frontiers in Artificial Intelligence and Applications, pp.3-12, IOS Press.

Buitelaar, P., Olejnik, D., and Sintek, M. 2004. A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis, in *Proc. of the 1st European Semantic Web Symposium*, pp.31-44, Heraklion.

Cañas, A. J., Hill, G., Carff, R., Suri, N., Lott, J., Eskridge, T., Gómez, G., Arroyo, M., & Carvajal, R. 2004. In: Concept Maps: Theory, Methodology, Technology, in *Proc. of the First International Conference on Concept Mapping*, A.J. Cañas, J.D. Novak, and F.M. González (Eds.), pp. 125-133, Pamplona, Spain.

De Marneffe, M-C., MacCartney, B. and Manning, C.D. 2006. Generating Typed Dependency Parses from Phrase Structure Parses, in *Proc. of 5th Conference on Language Resources and Evaluation LREC-06*, Genoa.

Haase, P., and Volker, J., 2005. Ontology Learning and Reasoning - Dealing with Uncertainty and Inconsistency. In Paulo C.G. da Costa, Kathryn B. Laskey, Kenneth J. Laskey, Michael Pool (Eds.): *Proc. of the Workshop on Uncertainty Reasoning for the Semantic Web*, pp. 45-55.

Frank, E., Paynter, G.W., Witten, I..H., Gutwin, C., Nevill-Manning, C.G., 1999. Domain-specific key phrase extraction, in *Proc. of the 16th International Joint Conference on Artificial Intelligence*, pp. 668-673, Morgan Kaufmann Publishers, San Francisco.

Hwang, C.H. 1999. Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information, in *Proc. of the 6th International Workshop on Knowledge Representation meets Databases*, pp. 14-20, Linkoeping, Sweden.

Klein, D. and Manning, C.D., 2003. Accurate unlexicalized parsing, in *Proc. of the 41st Meeting of the Association for Computational Linguistics,* pp. 423 – 430, Sapporo, Japan.

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 (FLAIRS 2006) Special Track on Intelligent Tutoring Systems*, Melbourne Beach, FL.

Kumar, A., and Kahle, D.J. 2006. VUE: A concept mapping tool for digital content. Concept Maps: Theory, Methodology, Technology, in *Proc. of the Second International Conference on Concept Mapping,* A. J. Cañas, J. D. Novak, (Eds.), San José, Costa Rica, 2006.

Maedche, A., and Staab, S. 2000. Semi-automatic Engineering of Ontologies from Text, in *Proc. of the 12th International Conference on Software Engineering and Knowledge Engineering*.

Murray, T. 2003. An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art, in Murray, T., Blessing, S. and Ainsworth, S. (Eds.), *Authoring Tools for Advanced Technology Learning Environments. Toward cost-effective, adaptative, interactive, and intelligent educational software*, pp. 491-544, Kluwer Publishers.

Navigli, R., Velardi, P., and Gangemi, A. 2003. Ontology Learning and its application to automated terminology translation, *IEEE Intelligent Systems*, 18(1):22-31.

Nkambou, R., Frasson, C., Gauthier, G. 2003. CREAM-Tools: An Authoring Environment for Knowledge Engineering in Intelligent Tutoring Systems, in Murray, T., Blessing, S. and Ainsworth, S. (Eds): *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective, adaptative, interactive, and intelligent educational software*, pp.93-138, Kluwer Publishers.

Novak, J. D. and Cañas A. J. 2006. The Theory Underlying Concept Maps and How to Construct Them, *Technical Report IHMC CmapTools 2006-01*, Florida Institute for Human and Machine Cognition, 2006.

Park, Y. 2004. GlossOnt: A Concept-focused Ontology Building Tool, in *Proc. of KR*, pp. 498-506, Whistler.

Schutz, A. and Buitelaar, P. 2005. RelExt: A Tool for Relation Extraction from Text in Ontology Extension, in *Proc. of the International Semantic Web Conference ISWC*, pp. 593–606.

TMAPI 2006. Retrieved from : http://tmapi.org/

UIMA 2006. Retrieved from: http://www.research.ibm.com/UIMA/.

Wikipedia, 2006. Retrieved from: http://en.wikipedia.org/wiki/Concept_map.

Zouaq, A., Frasson, C., and Rouane, K. 2000. The Explanation Agent, in *Proc. of the 5th International Conference on Intelligent Tutoring Systems*, pp. 554-563, Montreal, Lectures Notes in Computer Science, Springer Verlag nº 1839.