

Building Domain Ontologies from Text for Educational Purposes

Amal Zouaq and Roger Nkambou, *Member, IEEE*

Abstract—This paper presents a semiautomatic framework that aims to produce domain concept maps from text and then to derive domain ontologies from these concept maps. This methodology particularly targets the e-learning and Artificial Intelligence in Education (AIED) communities as they need such structures to sustain the production of e-learning resources tailored to learners' needs. This paper details the steps to transform textual resources, particularly textual learning objects (LOs), into domain concept maps, and it explains how this abstract structure is transformed into a formal domain ontology. A methodology is also presented to evaluate the results of ontology learning. The paper shows how such structures (domain concept maps and formal ontologies) make it possible to bridge the gap between e-learning and Intelligent Tutoring Systems by providing a common domain model.

Index Terms—Applications and expert knowledge-intensive systems (education), intelligent Web services and Semantic Web knowledge acquisition, knowledge reuse.

1 INTRODUCTION

THE importance of automatic methods to enrich knowledge bases from free text is acknowledged by the knowledge management and ontology communities. Developing a domain knowledge base is an expensive and time-consuming task, and static knowledge bases are difficult to maintain. This is especially true in the domain of online training. Generally split into e-learning, adaptive educational hypermedia, and Intelligent Tutoring System (ITS) communities, the online educational community lacks common views, methods, and resources to build a knowledge base [10]. In fact, integration and cooperation between the e-learning and the ITS communities can only benefit all groups. On one hand, e-learning-based environments focus on the reusability of learning resources. However these resources are not adaptable to suit learners' needs, they fail to use explicitly stated instructional strategies, and they lack rich knowledge representations. On the other hand, ITSs exploit rich knowledge structures, provide adaptive feedback, and implement pedagogical strategies. However, their knowledge base is not generally reusable as it is dependent on the application domain and proprietary programming. If we consider learning objects (LOs) as resources to semi-automatically build ITS knowledge bases, then it should be possible to exploit the benefits of both worlds. In fact, various studies have requested that LOs be enriched semantically [21], [26], [31], as well as announcing the creation of the educational Semantic Web [4]. The LORNET project [29], [38], a significant Canadian initiative to reach this goal, aims at providing Semantic Web applications for e-learning and Knowledge Management systems.

Along the same line of research, the Knowledge Puzzle Project proposes building a knowledge base that is usable by both communities (ITS and e-learning). A domain ontology is central to this knowledge base. Other required structures include the learning objective and instructional strategy specifications. This paper focuses mainly on the domain model and describes a semiautomatic methodology and tool, TEXCOMON, to build domain ontologies from English text.

One of the distinctive features of TEXCOMON's approach lies in the use of intermediate knowledge models (concept maps) to generate the domain ontology. Since this ontology is dedicated to education, we propose the use of concept maps as intermediate structures. Concept maps tend to make the structure of a body of knowledge much more significant for human users than other forms of knowledge representation [35]. Hence, they are more easily validated and enriched by a domain expert. Concept maps also foster meaningful learning and index sentences at a fine-grained level, which is required for efficient LO indexing and retrieval. In order to promote interoperability and reuse, concept maps pass through an export process that outputs a lightweight domain ontology.

Building ontology learning frameworks still requires the supervision of a domain expert. To validate the resulting ontology, this paper also describes an evaluation methodology and presents the results obtained in an evaluation with a corpus of text documents. The Knowledge Puzzle offers a series of services that can be used in both e-learning and ITS environments. Such services are used to exploit the generated knowledge base for computer-based training.

The paper is organized as follows: First, related work is presented (Section 2) before the philosophical foundations of the Knowledge Puzzle are described and certain definitions are provided (Section 3). Second, the semiautomatic methodology for knowledge acquisition from text is described, as are the domain concept maps and the generation of ontologies (Section 4). Third, we present an

• The authors are with the University of Quebec at Montreal, Pavillon Sherbrooke 200, rue Sherbrooke ouest, local SH-5720, Montréal, QC H2X 3P2, Canada. E-mail: {zouaq.amal, nkambou.roger}@uqam.ca.

Manuscript received 22 Mar. 2008; revised 4 July 2008; accepted 16 July 2008; published online 30 July 2008.

For information on obtaining reprints of this article, please send e-mail to: tlit@computer.org, and reference IEEECS Log Number TLTSI-2008-03-0032. Digital Object Identifier no. 10.1109/TLT.2008.12.

approach to evaluate the ontology and perform comparative analyses with Text-To-Onto [28] (Section 5). Fourth, the value of the approach for the educational community is highlighted by demonstrating how the generated ontology can sustain a knowledge base that offers a series of services to the Artificial Intelligence in Education (AIED) community (Section 6). Finally, conclusions are drawn.

2 RELATED WORKS

This section presents related work on concept maps and domain ontology generation. It also describes how domain ontologies enrich LOs.

2.1 Generating Concept Maps

Concept maps are a valuable resource whose rich structure can be exploited to retrieve information and train learners. This paper introduces a solution to semiautomatically generate concept maps from domain documents. These concept maps are useful to support meaningful learning and serve both as a knowledge base for building domain ontologies and as a skeleton for composing more detailed LOs. Knowledge extraction is based on lexico-syntactic and semantic patterns. Previous and concurrent studies have also attempted to generate concept maps from documents [13], [49]. The main difference with Knowledge Puzzle's approach is that such studies do not attempt to convert concept maps into domain ontologies. Furthermore, they fail to use other nonverbal forms of knowledge, such as prepositional forms to link phrases, unlike the Knowledge Puzzle. Our approach provides for the progressive construction of concept maps with meaningful domain concepts. In this novel proposal, entire sentences are exploited in order to find as many semantic relationships as possible, whereas other approaches only exploit verbal relationships.

Automatic authoring efforts have also been made in the area of adaptive hypermedia systems [4], [5], [15]. Such efforts include adding missing attributes in the domain model by performing link analysis [5] and creating new links using some metrics such as relatedness calculations [15]. However, these efforts did not try to make concept maps emerge from text, which is of major importance for automatic indexing.

2.2 Generating Domain Ontology

Generating and populating ontologies from text are two very active research fields. Related projects include: Mo'k [8], Text-2-Onto [22], OntoLT [12], KnowItAll [18], TEXTRUNNER [5], OntoGen [19], SnowBall [1], and OntoLearn [34]. Some studies attempt to handle the entire process of knowledge acquisition (concept, instance, attribute, relationship, and taxonomy), while others only address certain segments of it, using methods such as statistical analysis, linguistic parsing, Web mining, and clustering. A good review on ontology learning from text can be found in [11]. Several projects now use machine learning techniques to generalize, classify, or learn new extraction patterns (i.e., KnowItAll, TEXTRUNNER, OntoGen, SnowBall, and OntoLearn) without necessarily resorting to linguistic analyses. Some combine two approaches,

such as Text-To-Onto or Mo'K, which specializes in the learning of conceptual categories.

Overall, very few approaches have used concept maps to generate domain ontologies or as a layer to index textual LOs. Also, very few investigations from the AIED community attempt to handle the issue of automatically managing textual documents to improve indexing and retrieval. This work is an effort in this direction, and an interesting and recent study, the Language Technology for e-learning Project [30], further envisions the use of multilingual language technology tools and Semantic Web techniques to improve the retrieval of learning material.

2.3 Learning Object Semantics and Services

Providing semantic-rich learning environments is one essential issue in current computer-based education. As stated previously, this is a line of research that is being pursued by a number of efforts, including [26], [31], [37], and [50]. The work presented in [26] and [48] underlines the importance of instructional roles and context in building LOs. This context includes domain ontology, instructional role, and instructional design management. However, no automatic methods for building ontologies are provided for such contexts. This is a significant limitation, given the considerable effort required by designers who articulate the domain knowledge. Moreover, from the beginning, the instructional design is restricted to a single methodology (e.g., IMS-LD), which reduces the ability to benefit from the proposed solution in non-IMS-LD platforms. A solution has yet to be provided to move to other standards or learning environments. Thus, it is difficult to anticipate the interest of such solutions in ITS. Research conducted by the LORNET network [37], [38] cites the importance of providing LOs with semantics, although their solution is merely offering a set of authoring tools for inputting such content manually. To the authors' knowledge, this paper presents the first initiative to exploit semiautomatic ontology learning techniques in order to express LO semantics.

3 THE KNOWLEDGE PUZZLE APPROACH: FOUNDATIONS

The Semantic Web vision relies on domain ontologies to describe Web content and make it understandable by software agents. Computer-based educators, particularly those from the e-learning community, realize the importance of this vision to sustain the production of reusable LOs [21], [26]. The relevance of domain ontologies has also grown in ITSs, and its usefulness to model expert and learner models is recognized [44]. On the whole, new generations of robust ontology engineering environments such as Protégé [41] have fostered the creation of ontologies. In such a context, the use of domain ontologies to bridge the e-learning, AIED, and ITS communities seems to be a promising solution for a number of issues pertaining to domain knowledge acquisition and dissemination through computer-based education. Since knowledge acquisition bottlenecks represent the worst problem for knowledge-based systems, it is important to explore semiautomatic

methods to generate ontologies. First, however, several theoretical issues must be considered.

First issue: can we build domain ontologies from text?

We postulate that most conceptual and terminological domain structures are described in documents. Thus, applying an ontology generated from texts seems to be a promising avenue of study. Creating ontology-based metadata that can be understood by machines reflects the vision of the Semantic Web. Semantic languages such as RDF and OWL are used in order to express semantic annotations in a standard way. Therefore, this paper aims at using textual LOs from a given domain as input for a knowledge acquisition process.

Second issue: what kind of knowledge should we extract? Educational resources predominantly focus on two types of knowledge: declarative and procedural knowledge. Since ontologies basically represent declarative knowledge, this paper concentrates on such statements. From our perspective, procedural knowledge is best represented through rules and does not belong in a domain ontology. In fact, as described by the Semantic Web architecture, the rule layer is on top of the ontology layer [7]. Another important question is: should ontology acquisition tools be able to produce consistent ontologies from texts of different fields? As answering this question pertains to a long-term issue, the paper focuses on techniques to extract consistent domain ontologies from documents pertaining to a single domain (here, the SCORM domain).

Third issue: should educational ontologies be generated through approaches that differ from those used in other domains? To answer this question, sources of knowledge representation were investigated, namely, general network-based representations and specific semantic networks. Due to their human-centered origins and their proximity to the field of lexical acquisition, mapping text content and semantic networks somehow seemed natural. In fact, we believe that semantic networks or concept maps consist of interesting and expressive knowledge models that represent learning content. However, semantic networks suffer from an inherent semantic ambiguity. For example, we were unable to differentiate individuals from concepts in the resulting concept maps. Moreover, due to the direct translation of written sentences into concept map sentences, various terms were used to express synonyms, resulting in further ambiguity. In order to reason consistently, another representation was required to effectively represent the ontological content rather than only the learning content: we needed a mining process over the generated concept maps in order to detect ontological concepts and relationships. Description logic (DL), a subclass of first-order logic, appeared to be the most suitable option to formalize concept maps. As an offspring of semantic network representations, DL was able to adequately represent the resulting knowledge and provide inference capabilities. However, another bridge was required between the concept maps and the domain ontology. More precisely, it was decided to provide formal semantics to concept maps through graph theories. These issues are further described in the remainder of the paper.

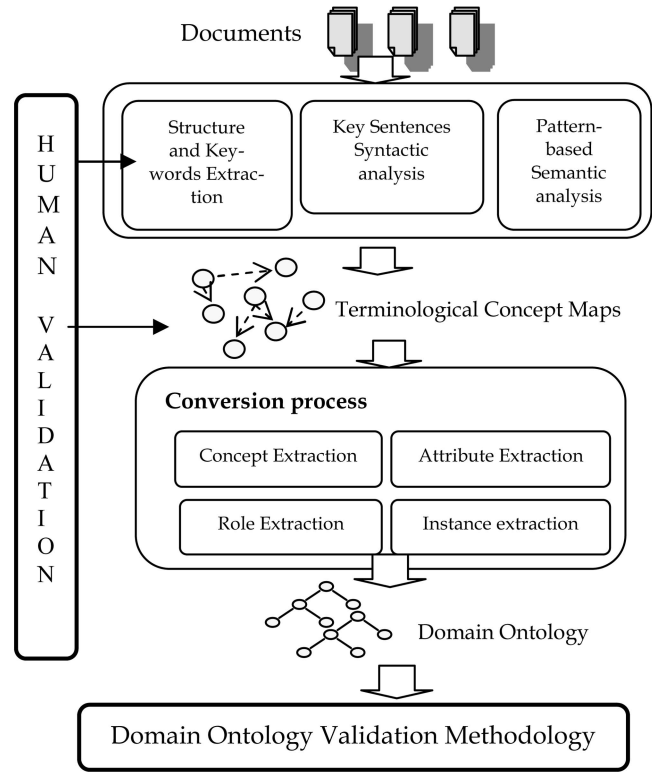


Fig. 1. The Domain Knowledge Acquisition Process.

Finally, an additional concern: the possibility of reusing an approach first dedicated to education in other disciplines. In other words, we wondered if the TEXCOMON process was applicable to any other domain. As shown in Section 6, this was successfully demonstrated by comparing our approach with a state-of-the-art ontology learning tool: Text-To-Onto [32].

4 THE KNOWLEDGE ACQUISITION PROCESS THROUGH TEXCOMON

In computer-based education, particularly in the field of ITSs, domain knowledge is defined as representations of expert knowledge. It is assumed that such representations can be expressed through concept maps [33], [36].

TEXCOMON stands for TEXT-Concept Map-ONtology to indicate the process followed in order to convert texts into domain concept maps, which are in turn transformed into an OWL ontology. This ontology represents the implicit domain knowledge contained in LOs, which has yet to be made accessible in training environments. Fig. 1 shows the domain knowledge acquisition process.

Human validation is essential for each step of the knowledge acquisition process. Designing domain ontologies does not follow a linear process: it involves numerous revisions before a final consensual solution is developed. Moreover, as the Knowledge Puzzle platform is designed with the ultimate goal of training, it is important to validate the results obtained at each step of the mining process. Humans should confirm and complete the results to guarantee the quality of the ontology.

The learning process in the Knowledge Puzzle's approach is syntactic and domain independent. The Knowledge

Puzzle instantiates a series of extraction rules from a set of domain-independent templates. Contrary to many other ontology learning approaches, it does not rely on a supervised method to guide the learning process (i.e., by providing examples and learning a model).

As shown in Fig. 1, the domain knowledge acquisition process relies on a number of steps. The first step involves extracting the document structure and mining domain terms and relationships. This results in terminological concept maps. The second part of the process involves the conversion of concept maps into a domain ontology by detecting classes, associations, attributes, and instances and saving them in OWL. Validation is performed by a human expert at each stage of the process, especially to assess the correctness of the generated keywords and concept maps within the TEXCOMON environment. The domain ontology validation methodology is explained further in Section 5.

4.1 Detecting Document Structure and Keywords

Detecting the document structure is not trivial due to the multitude of available formats (txt, doc, pdf, html, etc.). Moreover, this issue cannot be avoided when it comes time to analyze documents and, more specifically, to extract sentences to be parsed. Aside from sentences, there is a need to detect other components of LOs (paragraphs, images, tables, lists, examples, definitions, etc.) to face current e-learning challenges. For example, document structure should provide learners with the means to refer to the appropriate portion of a document that fulfills their current needs. This is also necessary for a dynamic LO composition, which must rely on more fine-tuned parts to fulfill a precise learning objective. Some studies have focused on structured documents such as XML or textbook documents, others have worked with pdf documents [17] or definition mining [33], but a complete framework able to handle the multitude of formats and structures is yet to be created.

The Knowledge Puzzle is restricted to plain text documents as it focuses on detecting and parsing documents at the sentence level and also since all mentioned formats (word, pdf, and html) can be transferred to plain text. Manual annotation capabilities were supplemented to enable additional structural annotations at both the pedagogical level (e.g., annotating definitions and explanations in texts) and the basic structure level (e.g., tables and images). A long-term goal consists of creating a library of structure extractors that work with various document formats.

In its initial version, TEXCOMON works with plain text documents that are automatically partitioned into sets of paragraphs, which are in turn composed of series of sentences. This is performed with annotators developed with the Unstructured Information Management Architecture (UIMA) [47].

The other issue worth considering before beginning a mining process pertains to the set of keywords or seed words from texts that are fed into the system. Generally, such keywords are provided by human experts [25]. In the TEXCOMON approach, a machine learning algorithm, Kea-3.0 [20], is used to extract representative n-grams from documents. The initial algorithm was slightly modified to process one document at a time, in order to avoid working with a collection of documents simultaneously.

TABLE 1

A Sentence Grammatical Map for the Sentence “An Asset Can Be Described with Asset Metadata to Allow for Search and Discovery within Online Repositories, Thereby Enhancing Opportunities of Reuse”

```
det(Asset-2, An-1)
nsubjpass(described-5, Asset-2)
aux(described-5, can-3)
auxpass(described-5, be-4)
nn(metadata-8, asset-7)
prep_with(described-5, metadata-8)
aux(allow-10, to-9)
purpcl(described-5, allow-10)
prep_for(allow-10, search-12)
conj_and(search-12, discovery-14)
amod(repositories-17, online-16)
prep_within(allow-10, repositories-17)
advmod(enhancing-20, thereby-19)
purpcl(described-5, enhancing-20)
dobj(enhancing-20, opportunities-21)
prep_for(enhancing-20, reuse-23)
```

The extracted key expressions (one or more words) are then used to collect sentences in which they are found. This enables domain terms and relationships to be extracted with respect to the detected keywords.

4.2 Key Sentences Syntactic Analysis

According to [14], two types of grammars represent the structure of sentences in natural languages: constituency grammars and dependency grammars. The selected grammar strongly influences the types of possible semantic analyses. Constituency grammars describe a phrase-structure syntax. In dependencies, each pair of word is related by a grammatical link called a dependency. From a knowledge representation perspective, dependency grammars have a major advantage when compared to constituency grammars: grammatical link dependencies are intuitively close to semantic relationships. Several analyzers can perform dependency analyses.

It is important that analyzers be as accurate as possible, i.e., able to produce accurate dependencies. The results presented in [44] suggest that the Stanford University Parser [28] can generate accurate analyses for most sentences encountered. In addition, research in natural language processing at Stanford University happens to be on the cutting edge of what is being done in the field today. For these reasons, the Stanford University Parser was used to transform sentences from documents into typed dependency representations.

Each sentence is represented as a Sentence Grammatical Map, i.e., a set of terms linked by the Parser's typed dependencies. As an example, Table 1 depicts a map that illustrates different grammatical dependencies.

TEXCOMON uses these grammatical maps in semantic analysis.

4.3 Pattern-Based Semantic Analysis

In this study, a pattern is represented through a set of input and output links. Such links represent the numerous grammatical relationships that are output by the dependency

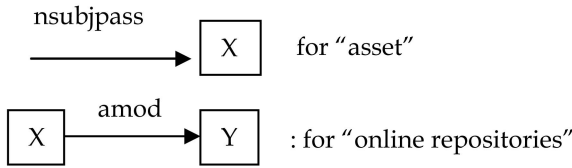
module [16] of the Stanford University Parser. Once a pattern is identified, a method is triggered to compute the semantic structure associated with it.

4.3.1 Extracting Terminology

Terminology extraction refers to the discovery of terms that become potential candidates for concepts in an ontology. It can be facilitated by the exploitation of LOs as the primary source of knowledge: LOs are purely didactic documents, providing definitions and explanations about the concepts to be learned. These concepts share the properties of low ambiguity and high specificity, due to their natural goals in the learning context.

A set of rules was established to exploit the grammatical maps so as to retrieve specific predefined patterns. These patterns are used to extract a Sentence Semantic Concept Map from the grammatical one (semantic terms and relationships).

Terminology patterns rely on elements such as adjectives or nouns to restrict the meaning of a modified noun, e.g., *is-a* (ITS, Tutoring System). They also constitute a very accurate heuristic to learn taxonomic relationships [29]. For example, extracting terminology patterns allows for defining the domain terms used in the previous example (Table 1), which are Asset, Asset metadata, Opportunities, Reuse, Search, Discovery, and Online repositories. Some of the patterns used in the example are



All of these terms become candidate terms to express domain concepts.

4.3.2 Extracting Relationships

Domain terms must be related in some ways. Extracting relationships refers to identifying linguistic relationships among the discovered terms. Verbs and auxiliaries, which generally also express domain knowledge, become central to such extraction. Again, grammatical pattern structures similar to the ones employed in the previous step are exploited to extract relationships of interest. A verbal relationship pattern used in the example is shown below:

```

nsubjpass(described - 5, Asset - 2)
aux(described - 5, can - 3)
auxpass(described - 5, be - 4)
nn(metadata - 8, asset - 7)
prep_with(described - 5, metadata - 8)
  
```

This pattern outputs the relationship “can be described with.” Overall, a set of around 20 patterns was identified. At this stage, it is important to understand that there is no filtering of the “important” domain terms. All the key sentences are parsed, and all the recognized patterns are instantiated. The filtering will be done at a later stage during the conversion into an OWL ontology.

Each sentence is associated with its main subject (the term *Asset*, from the previous example). The process described above is repeated for all of the selected sentences. Domain concept maps are generated around given concepts by listing all the relationships where the concepts appear as main subjects. Each domain concept has a domain concept map that describes it and that links it to various documents through relationships and other concepts. This is called its context.

It is then possible to retrieve a term or a particular relationship and to be automatically directed to the source sentence, paragraph, or document. This allows enhanced retrieval of the appropriate knowledge. Contexts are also used to provide synthesized views of a concept, which can be highly useful in e-learning. Finally, one important aspect regarding contexts is that they are not only made up of binary relationships, but they also include paths of information. Such paths are used to supplement, clarify, and expand initial relationships. For example, in the sentence “metadata is-used-to-search LOs within online repositories,” the relationship “within,” which links the concepts “LOs” and “online repositories,” provides additional details to the first relationship.

4.4 Generating Domain Ontologies from Domain Concept Maps

Domain concept maps act as skeletons to build domain ontologies. This process implies determining classes, associations, attributes, and instances.

4.4.1 Defining Classes

Extracting ontological primitive classes from concept maps is performed by detecting high-density components. In TEXCOMON, a domain term is considered a class if

1. it is the main topic of various sentences, thus being a frequent subject in the domain of interest, and
2. it is linked to other domain terms through semantic relationships.

Note that a single concept can be expressed in different manners in a text. TEXCOMON can recognize the base form of a concept through stemming. It uses a Java version of the Porter Stemmer [40] to produce the stem associated with each concept. For example, the words “stemmer,” “stemming,” and “stemmed” have the same root: “stem.” This is particularly useful as it allows for recognizing plural forms and certain conjugated verb tenses. Another way of expressing concepts is through abbreviations (e.g., “SCO” stands for “Sharable Content Object”). Although the Stanford University Parser outputs abbreviation links as typed dependencies, it is not always reliable. Hence, TEXCOMON implements an algorithm to identify correct abbreviations, which are stored as acronyms of the current concept and exported as equivalent classes, as shown below.

```

<owl:Class rdf:ID="runtime_environment">
  <rdfs:subClassOf rdf:resource="#environment" />
  <owl:equivalentClass>
    <owl:Class rdf:ID="RTE" />
  </owl:equivalentClass>
</owl:Class>
  
```

At the time this paper was submitted, TEXCOMON was not equipped to handle anaphora resolutions and could not process antecedents such as “reference model” and “the model” in the following sentences: “SCORM is a reference model [...]”. The model [...].”

4.4.2 Defining Associations

In DL, associations consist of properties within a specific domain and range. Each domain and range refers to a class. Verbal relationships express more specialized relationships, which are important in the domain. Basically, all verbal relationships between pairs of classes are considered to be ontological relationships.

The relationships generated include simple object properties such as

```
<owl:ObjectProperty rdf:ID="may_need">
  <rdf:domain rdf:resource="#training_resources" />
  <rdf:range rdf:resource="#metadata" />
</owl:ObjectProperty>
```

An object property can also take the shape of a union of classes in its range or domain. This happens when the same relationship (e.g., describes) is encountered between a concept (e.g., metadata) and many other concepts (e.g., content_objects or assets).

4.4.3 Defining Instances and Attributes

Extracting instances enables finding objects that are instances of a particular concept. Hearst [24] first talked about linguistic patterns to identify hyponyms (“is a kind of”). Particularly, the pattern “NP1 such as NP2, NP3, and NP4” expresses a hyponymous relationship. For instance, in the sentence “media such as text and images,” text and images are considered as instances of the concept “media.”

It is sometimes difficult to differentiate linguistic expressions revealing “instance-of” relationships from expressions that indicate subclass relationships. Suppose that NP1 represents a concept. TEXCOMON uses the following rules to establish whether a given link consists of a subclass link or an instance link:

- If NP2, NP3, and NP4 are also concepts, they are considered subclasses of NP1.
- Otherwise, if NP2, NP3, and NP4 are not considered concepts, they are stored as instances of NP1.

Obviously, the different instance patterns apply only to ontological classes. Examples of extracted instances include

```
<grouping rdf:ID="IMS" />
<grouping rdf:ID="ARIADNE" />
```

As far as attributes are concerned, they describe the concept itself. They can be extracted by using contextual information or relying on nominal modifiers to express potential properties. TEXCOMON uses the following patterns to extract concept attributes:

- < attr> <C> <verb> ..., where C denotes a concept, and attr denotes a modifier. A sample text that matches this pattern would be the following: ... inline metadata is ..., where metadata is a concept.

- < attr> of <C> (e.g., “identifier of asset”) or <C>’s< attr> (“asset’s identifier”).
- <C> have/possess < attr>.

Similar techniques to identify concept attributes are found in [3] and [39]. If < attr> is a concept, the attribute is considered an OWL Object Property; otherwise, it is created as a Data Type Property.

5 DOMAIN ONTOLOGY EVALUATION IN THE KNOWLEDGE PUZZLE

Increased use of domain ontologies requires well-established methods to evaluate them. This section investigates the performance of the domain ontology generated through TEXCOMON, based on a certain number of measures.

5.1 Evaluation Methodology

Evaluating ontologies remains an ongoing research challenge, and various methods have been proposed, as summarized in [9]. However, one possible criticism of such approaches is that they are only designed to evaluate certain specific characteristics of the ontology. It seems that ontology assessment is even more critical with methods generated automatically. Each extraction step must undergo both quantitative and qualitative evaluations, e.g., evaluation of terms, concepts, taxonomy, and conceptual relationships. For these reasons, a four-dimensional evaluation methodology is proposed:

A syntactic evaluation uses RACER-PRO [42] to assess the consistency of the ontology.

A structural evaluation strives to detect the structural characteristics of the generated domain ontology. Based on different measures, these characteristics can be helpful for ontology designers who must select the available ontology that best suits their needs.

A semantic evaluation involves human domain experts. They assess the quality of the ontology or, at least, the plausibility of its concepts and relationships.

A comparative evaluation juxtaposes results by running different state-of-the-art tools on the same corpus. Given that generating domain ontologies is not an exact science in terms of processes and results, one of the most interesting evaluation indicators in this field consists of testing and comparing the available ontology learning tools with the same corpora.

5.2 Experiment Description

We used a corpus about the SCORM standard compiled from two handbooks: SCORM 2004 Third Edition Content Aggregation Model (CAM) and SCORM 2004 Third Edition Runtime Environment [43]. We created 36 plain text documents from these two handbooks (approximately 29,879 words, 1,578 sentences, and 188 paragraphs) by manually excluding sample code and certain chunks such as “Refer to Section” and selecting declarative sentences. From this corpus, TEXCOMON extracted a set of 1,139 domain terms and 1,973 semantic relationships.

The essence of the experiment was to select a set of key terms from the SCORM domain in order to detect their characteristics in the generated ontology. The underlying

TABLE 2
Set of Domain Representing Terms Sought

Key Search Terms
Asset
SCO
SCORM Content Model
SCORM
LMS
Runtime Environment
Metadata
SCORM Content Packaging
Activity
Content Organization
API
PIF

assumption of such experiments is that ontologies are representative of search terms if [2]

1. the search terms exist as classes in the ontology,
2. there is a close structural proximity to the corresponding classes,
3. the corresponding classes are richly described,
4. the corresponding classes are interlinked through many relationships, and
5. the corresponding classes are central in the ontology.

Table 2 presents the sought terms considered key concepts for the SCORM domain. These terms have been validated by the domain experts as being representative of the SCORM domain. We do not pretend that this is an exhaustive list or a best set. However, the experts agreed that these keywords should normally exist in an ontology defining the SCORM standard.

The other facets of the experiment intended to test whether more or less compact ontologies affect the quality of the results. As stated above, a domain term is considered a concept when it is involved with other concepts in a number of output relationships. Such number of relationships can be parameterized. This experiment considers four values of the number of output relationships (I): $I = 2$ (KP-2), $I = 4$ (KP-4), $I = 6$ (KP-6), and $I = 8$ (KP-8).

Finally, seven corpuses from 36 documents were created and organized. Basically, each corpus was a superset of the previous one. This permits assessing the quality of the ontology as new domain documents are added to the previous corpus (Table 3). This also facilitates a better understanding of the contribution of certain specific documents to the ontology.

5.3 Syntactic Evaluation

As a knowledge representation module, the syntax and semantics of OWL ontologies must be validated. Reasoners are typically used to compute classes that cannot be satisfied, subsumption hierarchies, and individual types. TEXCOMON uses RACER PRO [42] in order to validate the consistency of the ontology.

TABLE 3
Corpus Description

Corpus	Number of files	Number of paragraphs	Number of sentences
Corpus 1	10	76	728
Corpus 2	14	85	781
Corpus 3	18	104	921
Corpus 4	22	121	1086
Corpus 5	26	144	1294
Corpus 6	30	169	1450
Corpus 7	36	188	1578

Concepts that cannot be satisfied signal faulty modeling. Discovering inconsistent concepts catches the attention of human validators who can subsequently correct them.

5.4 Structural Evaluation

The structural evaluation approach is based on a set of metrics (defined in [2]) that consider the ontology a graph entity. Initially, such metrics were developed to rank ontologies and sort them for retrieval purposes, similar to Google's PageRank algorithm. Given a set of search terms, Alani and Brewster [2] searched for the best ontology to represent these terms.

These four structural metrics consist of the Class Match Measure (CMM), the Density Measure (DEM), the Betweenness Measure (BEM), and, finally, the Semantic Similarity Measure (SSM). The total score of these four units is added to rank ontologies with respect to specific terms sought.

We implemented functions (ONTO-EVALUATOR library) to perform the different computations of the metrics based on the exact formulas described in [2]. The following sections describe all the metrics.

5.4.1 The Class Match Measure (CMM)

The CMM evaluates the coverage of an ontology for the provided keywords (Table 2). Given the input keywords, the ONTO-EVALUATOR searches through the ontology classes to determine if the keywords are expressed as classes (exact match) or if they are included in class labels (partial match).

Results show that CMM tends to improve as the threshold decreases within a corpus. This indicates that many concepts that contain the sought terms (partial or total match) are deleted as the threshold increases, thus eliminating relevant concepts (according to the domain expert) that, in fact, should have been preserved. For cases that considered exact and partial matches, KP-2 and KP-4 seem to be the best ontologies.

An interesting phenomenon occurs when considering solely exact matches (classes whose labels are identical to the sought term): different results are obtained. In this case, KP-2, KP-4, and KP-6 yield identical results with the richest corpus. However, KP-8 performs worse than the others. This indicates that the sought terms, considered key domain terms, are involved in up to seven relationships with other domain terms.

Considering exact and/or partial matches may affect other metrics. In fact, most results are presented according to the number of matched classes resulting from the CMM.

When the impact of an exact match is clearly identified, it is indicated in the following metrics.

5.4.2 The Density Measure (DEM)

The DEM expresses the degree of detail or the richness of the attributes of a given concept. It is assumed that a satisfactory representation of a concept must provide sufficient detail regarding its nature. Density measurements include the number of subclasses, inner attributes, and siblings, as well as the number of relationships maintained with other concepts.

The DEM tends to increase proportionally with the number of concepts. Such variations result from the abundant information in the new corpus. For example, in this experiment, Corporuses 6 and 7 contribute many new relationships, which explain a drastic DEM increase, especially when the threshold is 2.

5.4.3 The Betweenness Measure (BEM)

The BEM calculates the betweenness value for each sought term in the generated ontologies. It measures the extent to which a concept lies on the paths between others. Class centrality is considered important in ontologies. A high BEM shows the centrality of this class. As in ActiveRank [2], Onto-EVALUATOR uses the BEM provided by JUNG [23]. This algorithm calculates the number of shortest paths that pass through each concept in the ontology (considered a graph). A higher BEM is assigned to concepts that occur in a larger number of shortest paths between other concepts.

In this experiment, we noticed that a reasonable number of relationships must be retained in order to reach interesting BEMs. Again, thresholds 2 and 4 seem to generate the best results.

5.4.4 The Semantic Similarity Measure (SSM)

The last measure, the SSM, computes the proximity of the classes that match the sought keywords in the ontology. As Alani and Brewster [2] stated, if the sought terms are representative of the domain, the corresponding domain ontology should link them through relationships (taxonomic or object properties). Failure to do so may indicate a lack of cohesion in the representation of the domain knowledge.

The SSM is based on the shortest path that links a pair of concepts. The SSM never decreases, regardless of the threshold value. In general, high thresholds result in poorer performance of SSM values. However, with large corporuses, higher thresholds become more interesting.

As previously stated, considering solely exact matches has a greater impact on this metric. Exact matches lead to very similar results for KP-2, KP-4, and KP-6, especially with the richest corpus (Corpus 7), where identical results are obtained. This is not the case when both partial and exact matches are investigated.

Finally, based on these four metrics, an overall score is computed. Let $M = \{M[1], M[2], M[3], M[4]\} = \{CMM, DEM, SSM, BEM\}$, w_i be a weight factor, and O be the set of ontologies to rank. The score is computed as follows [2]:

$$Score(o \in O) = \sum_{i=1}^4 w_i \frac{M[i]}{\max_{1 \leq j \leq |O|} M[j]}.$$

Identical or different weights can be assigned to each metric. The overall score is further explained in the comparative evaluation, where TEXCOMON and Text-To-Onto ontology scores are presented.

5.5 Comparative Evaluation

Comparative evaluations are performed by juxtaposing overall scores of the TEXCOMON ontologies (KP-2, KP-4, KP-6, and KP-8) and those of the Text-To-Onto ontologies. To perform such comparisons, two ontologies (TTO-1 and TTO-2) are generated from each corpus with Text-To-Onto, a state-of-the-art tool in ontology learning from text. The main difference between these two ontologies resides in the use of a different support in association rule learning (respectively, 0 and 0.1).

To generate a domain ontology using Text-To-Onto, the latter was run on each of the seven corporuses using the KAON Workbench [32].

The following actions were performed for each corpus:

- *Term extractions.* These actions consist of using a number of statistical measures to identify domain terms found in texts.
- *Instance extractions.* These actions aim at populating the domain ontology.
- *Association rule extractions with a minimum support of 0 and another of 0.1.* Learned associations are added to the ontology as properties. Association rule learning endeavors to discover frequently co-occurring items within a data set to extract rules related to such items. The support of association rules equals the percentage of groups that contain all of the items listed in such association rules. This experiment shows that even the support of a 0.1 threshold discarded all association rules generated by Text-To-Onto. In fact, TTO-2 results are significantly disparate compared to those from TTO-1, which contains numerous meaningless properties that enhance the value of certain structural metrics. Hence, it was decided to present results for both ontologies, TTO-1 and TTO-2, although the Knowledge Puzzle Ontologies and TTO-2 are actually being compared.
- *Relation learning.* This action aims to extract verbal relationships.
- *Taxonomy learning.* This action, using the Taxo Builder tool, aims at discovering hierarchical relationships. The combination-based approach from Hearst's patterns and heuristics is used. The FCA-based approach is not exploited for two reasons: first, no comparison basis existed for the Knowledge Puzzle, and second, the authors were not strongly convinced by the results of the formal concept analysis (neither with verbs nor with lexicographer classes).

The Pruner and the OntoEnricher were not used. In fact, OntoEnricher is supposed to enhance the ontology by using other external resources such as Wordnet. However, we wanted to compare information extraction from the same corpus without resorting to other knowledge sources. The pruner was tested but actually suggested pruning concepts that should not be removed from the resulting ontology

TABLE 4

Results for Corpus 7: Equally Distributed Weights (0.25) for CMM, DEM, BEM, and SSM

Ontology	Score	Rank
KP-2	1	1
KP-4	0.41	2
KP-6	0.34	4
KP-8	0.26	6
TTO-1	0.38	3
TTO-2	0.29	5

(for example, adl initiative, activity tree, and content organization). We believe that this is because Text-To-Onto relies only on statistical features (cumulative frequency) to prune some concepts and tends to keep only statistically significant concepts.

The overall scores of Text-To-Onto and TEXCOMON ontologies are generated with the same metrics.

Overall scores. The total score is computed once the four measures are applied to all generated ontologies. The total score is calculated by adding all the measurement values, taking into account the weight of each measure, which can be adapted to reflect the relative importance of each value for ranking purposes. Scores are computed for ontologies generated from the entire corpus, namely, Corpus 7.

When using equally distributed weights (0.25) for all metrics (Table 4), it is clear that TEXCOMON outperforms Text-To-Onto (when compared to TTO-2). KP-8 is the only ontology whose score is lower than TTO-2.

Moreover, when considering scores from a CMM with exactly matching input terms, using Corpus 7 and a weight distribution of 0.5, 0, 0, and 0.5, we obtained the results shown in Table 5. Here, KP-4 has a better overall score than KP-2. This means that when two metrics, namely, CMM and SSM, are more important for designers, KP-4 would be the best ontology option.

One last example investigates the results of considering solely the CMM. Here, KP-2, KP-4, and KP-6 obtain identical scores and ranks (Table 6).

Another type of assessment, called semantic evaluation, aims at detecting to what degree and how well the generated domain ontology reflects the domain knowledge. We believe that such evaluation can only be performed by human domain experts.

TABLE 5

Results for Corpus 7: Different Weights for CMM, DEM, BEM, and SSM, Respectively, 0.5, 0, 0, and 0.5

Ontology	Score	Rank
KP-2	0.99	2
KP-4	1	1
KP-6	0.99	3
KP-8	0.84	4
TTO-1	0.62	5
TTO-2	0.46	6

TABLE 6

Results for Corpus 7: 100 Percent of the Weight on CMM

Ontology	Score	Rank
KP-2	1	1
KP-4	1	1
KP-6	1	1
KP-8	0.83	3
TTO-1	0.92	2
TTO-2	0.92	2

5.6 Semantic Evaluation

Semantic analyses rely on human experts to assess the validity of the ontology. This evaluation comes to reinforce the results of the previous evaluations (syntactic, comparative, and structural).

Table 7 summarizes the evaluation of the four ontologies KP-2, KP-4, KP-6, and KP-8, using the mean scores for pertinence, as expressed by two experts. These experts are specialized in the implementation of SCORM-compliant applications and the deployment of SCORM-based content. As shown in Table 7, such positive results are promising.

The same procedure is then repeated for the Text-To-Onto ontologies (Table 8). The “Pertinent Defined Classes” column disappears in Table 8 simply because Text-To-Onto does not extract any defined classes.

We can notice that the scores for pertinent primitive classes and hierarchical relationships are nearly identical: in fact, the only difference between TTO-1 and TTO-2 lies in the use of a different support in association rule learning. Moreover, TTO-1 generated a very important number of unlabeled relationships (5,683), among which only 18 relationships were considered pertinent by the two experts. The experts discarded the other relationships due to their very low support (< 0.1).

TABLE 7

Mean Scores for the Knowledge Puzzle Ontologies

Ontology	Pertinent primitive classes (%)	Pertinent defined Classes (%)	Pertinent hierarchical relationships (%)	Pertinent conceptual relationships (%)
KP-2	86.65	55.55	84.3	80.08
KP-4	90.84	100	84.83	89.65
KP-6	90	100	77.1	91.15
KP-8	90.32	100	75.28	93.12

TABLE 8

Mean Scores for the Text-to-Onto Ontologies

Ontology	Pertinent primitive Classes (%)	Pertinent hierarchical relationships (%)	Pertinent conceptual Relationships (%)
TTO-1	73.06	47.53	0.31
TTO-2	73.06	47.53	53.03

TABLE 9
Statistics Regarding Extracted Items

	KP-2	KP-4	KP-6	KP-8	TTO-1	TTO-2
Number of primitives classes	413	139	82	57	336	336
Number of Taxonomic Links	372	125	84	66	223	223
Number of non taxonomic links	288	153	103	74	5683	33

6 RESULT, ANALYSIS, AND DISCUSSION

Results of this experiment show that TEXCOMON ontologies obtained higher scores than the Text-To-Onto ontologies, especially when compared with TTO-2.

One interesting feature is the variation of weights and exact/partial matches and their impact on the overall ontology scores. The experiments suggest that partial matches can skew results and larger weights should be used with exact matches. Basically, some important questions must be taken into account in order to perform such variations:

- First, what are the most important metrics according to the domain, the goal(s), and the needs?
- Second and most important, given an ontology with low connectivity (KP-2), is it possible to obtain a more compact ontology and preserve the performance or scores of KP-2?

If the answer to the last question is affirmative, a more compact ontology should be favored over one that is less compact, as it includes more strongly interconnected concepts and still conserves the sought domain terms. For example, in Table 5, KP-4 must be selected, whereas Table 6 shows KP-6 to be the best ontology: it has the same score as KP-2 and KP-4, yet it is much more compact than both of them.

Table 9 compares the output of TEXCOMON and Text-To-Onto in terms of the number of concepts and relationships (taxonomic and nontaxonomic). Notice that

- TEXCOMON results can be parameterized, which is not the case for Text-To-Onto. Actually, ontology designers may be interested in larger or denser ontologies, and they should be given the opportunity for calibrating the generating process.
- The decreasing number of concepts and relationships in TEXCOMON is consistent with the threshold increase.
- An interesting aspect surfaces in the number of nontaxonomic links in TTO-2 (i.e., 33) compared to TTO-1 (i.e., 5,683). This drastic drop pertains to the 0.1 support used in TTO-2, meaning that TTO-1 created association rules with a support lower than 0.1. Such relationships contribute to a somewhat improved performance of TTO-1, especially for SSM measurements, although they actually have no “meaning” or real interest as ontological relationships. Moreover, the tremendous number

TABLE 10
Results for the Domain Term “SCO”

Number of	KP-2	KP-4	KP-6	KP-8	TTO-1	TTO-2
Super-classes	3	2	4	4	1	1
Sub-classes	2	2	2	2	2	2
Neighbors	6	0	0	0	2	2
Properties	18	12	10	7	118	0

TABLE 11
Results for the Domain Term “Asset”

Number of	KP-2	KP-4	KP-6	KP-8	TTO-1	TTO-2
Super-classes	3	2	2	2	1	1
Sub-classes	0	0	0	0	0	0
Neighbors	7	1	0	0	4	4
Properties	11	8	7	4	172	0

of extracted relationships makes the task difficult for ontology designers.

Another way of comparing both systems relates to the use of a keyword to observe results in the ontologies (KP and TTO). Again, significant differences appear between TTO-1 and TTO-2. Tables 10 and 11 illustrate such statistics for the terms “SCO” and “asset.”

TTO-1 reveals numerous properties (i.e., 118 properties in Table 10 and 172 in Table 11), especially when compared with TTO-2 (i.e., 0) and TEXCOMON. A closer look at the relationships actually generated shows an important number of noisy relationships.

Overall, Text-To-Onto presents two problems: first, it fails to extract relationship labels between concepts, especially relationship output by association learning, and second, it does not save the complete label of concepts and only stores the stem. TEXCOMON handles both types of labels (complete labels and stems) and also provides labels for relationships.

In general, TEXCOMON provides interesting results. For example, concepts are generally rich, and they have a sufficient quantity of parents (a multilevel ontology rather than a flat structure). It was also possible to generate some defined classes (by stating an equivalent class relationship between a concept and its acronym), which had not been done before. Additionally, conceptual relationship learning aspects are particularly interesting. Another interesting facet is that TEXCOMON offers the possibility of calibrating thresholds to suit ontology designers’ wishes.

Given a set of sought terms considered to be important domain concepts:

- Thresholds can be calibrated by emphasizing CMM if the most important feature is a partial or exact match of search terms as ontological concepts.
- If the important feature consists of having richly described concepts with an important number of attributes and relationships, the DEM should have a heavier weight in the overall evaluation.

- If the important feature targets richly interconnected concepts to make them central to the ontology, semantic similarities and betweenness should be favored.

We do believe that all measures are important. In general, when taking into account the overall scores, ontologies KP-2 and KP-4 seem satisfactory given the corpus sizes.

Although there is no single right way to evaluate ontologies, certain lessons can be drawn from these experiments:

1. In the absence of a gold standard for a particular domain ontology, it is not always possible to create one. Hence, another type of ontology assessment must be selected.
2. Comparing the generated domain ontology to those generated with state-of-the-art tools can be beneficial as this shows the added value of new tools or platforms. This confirms the interest for comparative evaluations as proposed in this paper.
3. Evaluating ontologies from a structural perspective can also be relevant, as shown in [2]. Comparing this structural evaluation with other generated ontologies, as conducted herein, is meaningful.

7 THE INTEREST OF THE APPROACH FOR THE AIED AND E-LEARNING COMMUNITY

The Knowledge Puzzle approach proposes the implementation of AIED techniques for LOs, which means capabilities for representing domain knowledge in LOs, demonstrating reasoning pertaining to this domain model, building instructional scenarios according to certain pedagogical theories, and adapting learning content to suit learners' needs. This can be achieved by using ontologies and exploiting a number of services offered to course designers, learners, and educational systems. The domain ontology is provided through the process described in this paper. The following sections illustrate the interest of the approach for the online educational community and underline how the domain ontology is exploited in an educational context.

In the context of the Knowledge Puzzle Project, we implemented an ontology-based Semantic Web Architecture that represents different models in the traditional ITS architecture through ontologies. Dynamic Learning Knowledge Objects (LKO) are generated and act as small tutoring systems: they are equipped with a domain model, a learner model, and a tutor model. These LKO exhibit various characteristics: they are active, domain knowledgeable, independent, reusable, and theory aware. We believe that this architecture pools a synergy of the strengths found in the fields of e-learning and ITSs. Such architecture is documented elsewhere [51], but the gist of the approach is briefly summarized here.

7.1 Providing Services for Course Designers

Given the tremendous amount of LOs, providing the means to annotate contents in a semiautomatic manner by generating domain ontologies is of the utmost importance. In the context of the Knowledge Puzzle Project, designers are provided with a series of tools to generate accurate

representations of LOs and their domain contexts. This semiautomatic annotation facility offers authoring support for course designers who can search through learning content more efficiently. However, domain knowledge is not the only aspect considered. In fact, certain studies highlight the importance of an instructional role ontology [48], [51] to define knowledge chunks in LOs that can be linked to domain concepts (for example, a definition for "SCORM"). These annotations, manually performed by the Knowledge Puzzle tools, can be recycled efficiently in other curricula that use such search facilities. Designers are also provided with the ability to state a learning objective as a competency to be mastered over domain concepts. Rule-based authoring tools are offered to link competency levels and instructional roles. Finally, the proposed platform enables the defining of rule-based instructional scenarios that exploit semantic annotations. These scenarios guide course designers in the process of creating learning resources, according to a chosen pedagogical theory.

7.2 Providing Services for Learners

The first service provided for learners comprises the ability to obtain LOs tailored to their needs and profile, which is an important issue in computer-based learning. This point is further explained in Section 7.3. A learner model is stored in an IMS ePortfolio to save information pertaining to the learners' prior knowledge and acquired competencies (which are linked to the domain model). This portfolio can then be imported by any training application that is compatible with the standard, including the Knowledge Puzzle.

The second service consists of using concept maps for constructivist learning: concept maps are not only considered intermediary templates to build the domain ontology. In fact, studies have revealed concept maps to be useful training assets in constructivist environments [30].

The Knowledge Puzzle defines a formal relationship between a concept and the generated concept map to enable reuse in training sessions. This formal relationship represents the concept context. This is very important within constructivist environments where prior knowledge is used as a framework to understand and acquire new knowledge.

The Knowledge Puzzle offers capabilities to explore learning material and related domain concepts. Moreover, as expert systems can justify their answers by providing inference execution traces, the Knowledge Puzzle can show textual contents, thus providing solid proof of its "expertise." This can be done at a more or less fine-grained level and related to the current concept, instance, or association.

7.3 Providing Services for Educational Systems

One of the most critical issues in current efforts toward semantic-rich learning environments pertains to offering services for dynamic LO composition given a set of parameters, including learners' profiles and competencies to be mastered.

The importance of better fine-tuned retrieval of learning content was brought up through the discussion of an ontology of instructional roles [48]. In fact, due to the very nature of their pedagogical goals, LOs include various types of instructional roles such as definitions, examples, and

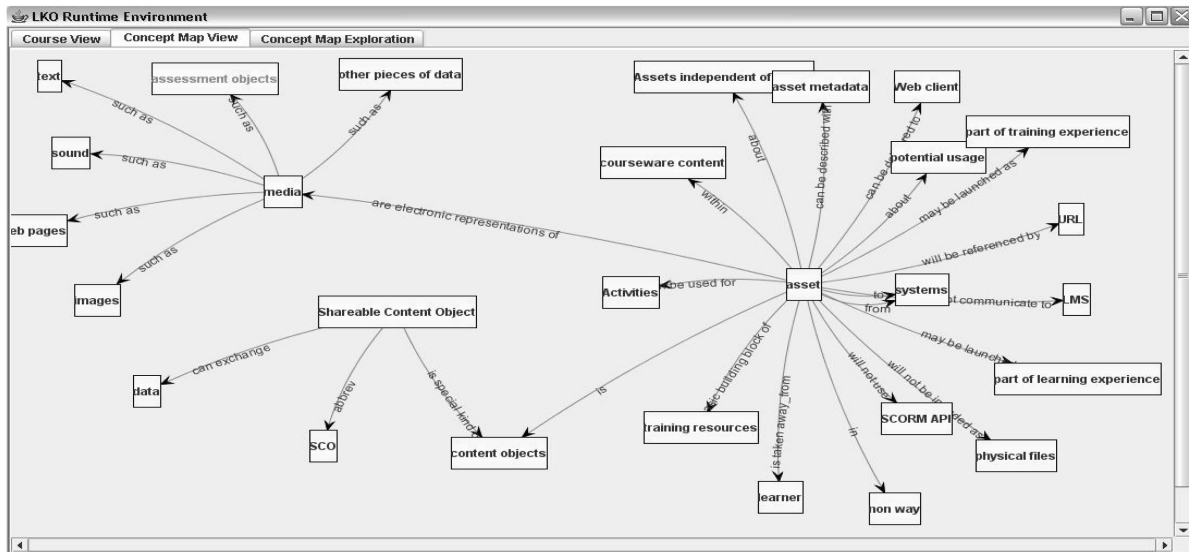


Fig. 2. A Concept Map Exploration in an LKO.

exercises. Thanks to the generated domain ontology, links between LOs and domain concepts are made explicit. Moreover, thanks to the instructional role ontology, links between instructional roles and LOs are also explicit. Finally, instructional roles are applied to domain concepts, thus providing an interrelated and meaningful structure (e.g., an explanation about concept X, a definition of concept Y, etc.). This structure is essential to offer composition services based on a given learner profile, the learning objective, and a specific instructional theory. The service automatically generates an adapted LO called LKO. As shown in Fig. 2, an LKO is encapsulated as an applet that contains a course view (with the planned curriculum) and a concept map view that indicates the various concepts and their links with one another.

Finally, another important service enables ontologies with robust semantics and languages to offer a certain level of reasoning for LO content. In fact, one of the main objectives when building a domain ontology is to provide systems with the ability to reason over this domain. However, to date, less attention was paid to this issue by the computer-based education community, which tends to focus more on knowledge representation facets. Reasoning involves a number of abilities considered by the Knowledge Puzzle:

The ability to query the ontology. The Knowledge Puzzle uses the Protégé OWL Java API and the SQWRL [46] query language and API to retrieve individuals or tuples that match a given query. Protégé also has the capability to use built-in SQWRL queries to examine the structure of an OWL ontology. This is helpful to reason over LO content by setting up high-level rules that retrieve

- concepts that are either more specific or more general,
- all the concepts related to a source concept X,
- properties that relate concept X to its context, and
- properties between concept X and concept Y.

The ability to reason over the ontology. In the context of the Knowledge Puzzle Project, reasoning over the ontology enables the discovery of related LOs or segments of LOs. It also makes it possible to create a pedagogical engine based on instructional theories [51] using rules defined in SWRL and the Jess Rule Engine. Other competencies include the abilities

- to explain the ontology, by using the source document from which a particular concept, association, or individual was created,
- to extend learner queries by using the domain ontology structure, and
- to cluster similar LOs according to their content.

Moreover, state-of-the-art DL reasoners such as RACER [42] exhibit satisfactory performance for knowledge bases with large TBOXES. This is particularly interesting for this study, as the main task of our learning tool entails the extraction of a TBOX rather than an ABOX from texts. The terms ABOX and TBOX are used to describe two different types of knowledge in ontologies. TBOX statements describe a system in terms of classes and properties. ABOX are TBOX-compliant statements about that conceptual schema. Since the object of this research is more on defining a conception about the domain of interest, the data mining task focuses on TBOX extraction.

Other current efforts include the option of providing automatic explanation facilities, as well as the ability of supplying answers to questions from texts, which is generally a challenging issue, even more so in the field of education.

8 CONCLUSIONS AND FURTHER WORKS

The Knowledge Puzzle Project proposes an entire framework to acquire and exploit knowledge in the fields of e-learning and ITSs. The proposed solutions for knowledge acquisition stems from a hybrid approach composed of natural language processing, pattern matching, and machine learning. On the one hand, it

contributes to the emergence of semantic-rich learning environments and borrows techniques from artificial intelligence to enrich LOs. On the other hand, it uses textual LOs or any textual document from a given domain as material to extract a relevant knowledge base for such domain. Given the difficulty of creating such a knowledge base, semiautomatic methods are essential to reduce the burden on experts, and they are necessary for ITSs. Furthermore, these methods enable the exploitation of LOs in ITSs. This opportunity equips ITSs to access learning material that, up to date, was devoted to traditional e-learning applications. In fact, this is an important issue due to the increasing number of LOs, and their wide adoption in the worlds of business and research.

Used as backbones to sustain entire frameworks, ontologies are dedicated to define the domain model, the expert model, the learner model, and the tutor model. Given the key role of ontologies in many applications, it is essential to provide tools and services to help users design and maintain high-quality ontologies. Automatic methods coupled with well-defined evaluation methods are crucial to their successful implementation in real-world settings. This paper presented a methodology to assess the quality of generated domain ontologies, and the results were compared with state-of-the-art tools in ontology learning from text. A corpus of text documents related to e-learning standards was used to illustrate this effort. Moreover, the interest of the approach also resides in services that can be offered to designers, learners, and educational systems, in order to exploit this knowledge base. These services include an on-the-fly composition process for LKOs, which differ from traditional LOs: they are generated according to a specific instructional theory, learner profile, and competence-based learning objectives, and they are enriched by domain structures that explain their content. Other services comprise authoring assistance and guidance components for course designers, as well as search and reasoning facilities for the domain ontology.

A number of enhancements and extensions are possible. We would like to enrich the pattern knowledge base with new structures and explore other ways of expressing patterns. Moreover, further thorough ontology and concept map evaluation techniques must be performed. Additionally, the increasing number of available ontologies raises concerns for their alignment and updating. At this stage of the project, automatically updating an ontology necessitates the execution of the text mining process over an enriched corpus. This execution creates a new OWL file that takes into account the enriched corpus. Future efforts should be directed at updating the existing ontologies by only inserting new ontological objects and checking the consistency of the modified ontology.

REFERENCES

- [1] Y. Agichtein and S. Gravano, "Snowball: Extracting Relations from Large Plain-Text Collections," *Proc. Fifth ACM Int'l Conf. Digital Libraries (DL '00)*, pp. 85-94, 2000.
- [2] H. Alani and C. Brewster, "Metrics for Ranking Ontologies," *Proc. Fourth Int'l Workshop Evaluation of Ontologies for the Web, 15th Int'l World Wide Web Conf. (WWW)*, 2006.
- [3] A. Almuhereb and M. Poesio, "Finding Concept Attributes in the Web Using a Parser," *Proc. Corpus Linguistics Conf.*, 2005.
- [4] L. Aroyo and D. Dicheva, "The New Challenges for E-Learning: The Educational Semantic Web," *Educational Technology & Soc.*, vol. 7, no. 4, pp. 59-69, 2004.
- [5] L. Aroyo, A.I. Cristea, and D. Dicheva, "A Layered Approach towards Domain Authoring Support," *Proc. Int'l Conf. Artificial Intelligence (ICAI)*, 2002.
- [6] M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open Information Extraction from the Web," *Proc. 20th Int'l Joint Conf. Artificial Intelligence (IJCAI '07)*, pp. 2670-2676, 2007.
- [7] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific Am. Magazine*, vol. 5, no. 1, pp. 34-43, 2001.
- [8] G. Bisson, C. Nédellec, and L. Canamero, "Designing Clustering Methods for Ontology Building—The Mo'K Workbench," *Proc. ECAI Ontology Learning Workshop*, pp. 13-19, CEUR-WS.org, 2000.
- [9] J. Brank, M. Grobelnik, and D. Mladenić, "A Survey of Ontology Evaluation Techniques," *Proc. Conf. Data Mining and Data Warehouses (SiKDD)*, 2005.
- [10] C. Brooks, J. Greer, E. Melis, and C. Ullrich, "Combining ITS and eLearning Technologies: Opportunities and Challenges," *Proc. Eighth Int'l Conf. Intelligent Tutoring Systems (ITS '06)*, pp. 278-287, 2006.
- [11] P. Buitelaar, P. Cimiano, and B. Magnini, "Ontology Learning from Text: An Overview," *Ontology Learning from Text: Methods, Evaluation and Applications*, P. Buitelaar, P. Cimiano, and B. Magnini, eds., vol. 123, Frontiers in Artificial Intelligence and Applications, pp. 3-12, IOS Press, 2005.
- [12] P. Buitelaar, D. Olejnik, and M. Sintek, "A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis," *Proc. First European Semantic Web Symp. (ESWS '04)*, pp. 31-44, 2004.
- [13] R.B. Clariana and R. Koul, "A Computer-Based Approach for Translating Text into Concept Map-Like Representations," *Proc. First Int'l Conf. Concept Mapping (CMC '04)*, vol. 1, pp. 125-133, 2004.
- [14] M.A. Covington, "A Fundamental Algorithm for Dependency Parsing," *Proc. 39th Ann. ACM Southeast Conf. (ACM-SE '01)*, pp. 95-102, 2001.
- [15] A. Cristea, "Is Semi-Automatic Authoring of Adaptive Educational Hypermedia Possible," *Advanced Technology for Learning*, vol. 1, no. 4, 2004.
- [16] M.-C. De Marneffe, B. MacCartney, and C.D. Manning, "Generating Typed Dependency Parses from Phrase Structure Parses," *Proc. Fifth Int'l Conf. Language Resources and Evaluation (LREC '06)*, pp. 449-454, 2006.
- [17] C. Desmoulin and M. Grandbastien, "Ontologies pour la Conception de Manuels de Formation à Partir de Documents Techniques," *Sciences et Techniques Educatives*, vol. 9, nos. 3-4, pp. 291-340, 2002.
- [18] O. Etzioni et al., "WebScale Information Extraction in KnowItAll (Preliminary Results)," *Proc. 13th Int'l Conf. World Wide Web (WWW '04)*, pp. 100-110, 2004.
- [19] B. Fortuna, M. Grobelnik, and D. Mladenić, "Background Knowledge for Ontology Construction," *Proc. 15th Int'l Conf. World Wide Web (WWW '06)*, pp. 949-950, 2006.
- [20] E. Frank, G.W. Paynter, I.H. Witten, C. Gutwin, and C.G. Nevill-Manning, "Domain-Specific Key Phrase Extraction," *Proc. 16th Int'l Joint Conf. Artificial Intelligence (IJCAI '99)*, pp. 668-673, 1999.
- [21] D. Gasevic, J. Jovanović, and V. Devedzic, "Ontologies for Creating Learning Object Content," *Proc. Eighth Int'l Conf. Knowledge-Based Intelligent Information and Eng. Systems (KES '04)*, pp. 284-291, 2004.
- [22] P. Haase and J. Volker, "Ontology Learning and Reasoning—Dealing with Uncertainty and Inconsistency," *Proc. ISWC Workshop Uncertainty Reasoning for the Semantic Web (URSW '05)*, pp. 45-55, 2005.
- [23] P. Hayes, T. Eskridge, R. Saavedra, T. Reichherzer, M. Mehrotra, and D. Bobrovnikoff, "Collaborative Knowledge Capture in Ontologies," *Proc. Third Int'l Conf. Knowledge Capture (K-CAP '05)*, pp. 99-106, 2005.
- [24] M.A. Hearst, "Automatic Acquisition of Hyponyms from Large Text Corpora," *Proc. 14th Int'l Conf. Computational Linguistics (COLING '92)*, pp. 539-545, 1992.

- [25] C.H. Hwang, "Incompletely and Imprecisely Speaking: Using Dynamic Ontologies for Representing and Retrieving Information," *Proc. Sixth Int'l Workshop Knowledge Representation Meets Databases (KRDB '99)*, pp. 14-20, 1999.
- [26] J. Jovanović, D. Gašević, C. Knight, and G. Richards, "Ontologies for Effective Use of Context in E-Learning Settings," *Educational Technology & Soc.*, vol. 10, no. 3, pp. 47-59, 2007.
- [27] JUNG, <http://jung.sourceforge.net/applet/index.html>, June 2007.
- [28] D. Klein and C.D. Manning, "Accurate Unlexicalized Parsing," *Proc. 41st Meeting of the Assoc. for Computational Linguistics (ACL '03)*, pp. 423-430, 2003.
- [29] LORNET, <http://www.lornet.org/>, Mar. 2008.
- [30] LT4eL, <http://www.lt4el.eu/>, Feb. 2008.
- [31] LUISA, <http://luisa.atosorigin.es/www/>, Mar. 2008.
- [32] A. Maedche and S. Staab, "Ontology Learning for the Semantic Web," *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 72-79, 2001.
- [33] S. Muresan and J.K. Klavans, "A Method for Automatically Building and Evaluating Dictionary Resources," *Proc. Third Int'l Conf. Language Resources and Evaluation (LREC '02)*, pp. 231-234, 2002.
- [34] R. Navigli and P. Velardi, "Learning Domain Ontologies from Document Warehouses and Dedicated Websites," *Computational Linguistics*, vol. 30, no. 2, pp. 151-179, 2004.
- [35] J.T. Nosek and I. Roth, "A Comparison of Formal Knowledge Representation Schemes as Communication Tools: Predicate Logic versus Semantic Network," *Int'l J. Man-Machine Studies*, vol. 33, pp. 227-239, 1990.
- [36] J.D. Novak and A.J. Cañas, "The Theory Underlying Concept Maps and How to Construct Them," Technical Report IHMC CmapTools 2006-01, Florida Inst. for Human and Machine Cognition, 2006.
- [37] G. Paquette, "An Ontology and a Software Framework for Competency Modeling and Management," *Educational Technology & Soc.*, vol. 10, no. 3, pp. 1-21, 2007.
- [38] G. Paquette and I. Rosca, "An Ontology-Based Referencing of Actors, Operations and Resources in eLearning Systems," *Proc. Second Int'l Workshop Applications of Semantic Web Technologies for E-Learning (SW-EL)*, 2004.
- [39] M. Poesio and A. Almuhareb, "Identifying Concept Attributes Using a Classifier," *Proc. ACL Workshop Deep Lexical Acquisition*, pp. 18-27, 2005.
- [40] M.F. Porter, "An Algorithm for Suffix Stripping," *Program*, vol. 14, no. 3, pp. 130-137, 1980.
- [41] *The Protégé Ontology Editor*, <http://protege.stanford.edu/>, July 2008.
- [42] RACER, <http://www.racer-systems.com/site/index.phtml>, Nov. 2007.
- [43] SCORM, <http://www.adlnet.gov/scorm/index.cfm>, Apr. 2007.
- [44] M. Stevenson and M.A. Greenwood, "Comparing Information Extraction Pattern Models," *Proc. ACL Workshop Information Extraction beyond the Document*, pp. 12-19, 2006.
- [45] P. Suraweera, A. Mitrovic, and B. Martin, "A Knowledge Acquisition System for Constraint-Based Intelligent Tutoring Systems," *Proc. 12th Int'l Conf. Artificial Intelligence in Education (AIED '05)*, pp. 638-645, 2005.
- [46] SQWRL, <http://protege.cim3.net/cgi-bin/wiki.pl?SQWRL>, June 2008.
- [47] UIMA, <http://uima-framework.sourceforge.net/>, Apr. 2007.
- [48] C. Ullrich, "Description of an Instructional Ontology and Its Application in Web Services for Education," *Proc. Workshop Applications of Semantic Web Technologies for E-Learning (SW-EL '04)*, pp. 17-23, 2004.
- [49] A. Valerio and D. Leake, "Jump-Starting Concept Map Construction with Knowledge Extracted from Documents," *Proc. Second Int'l Conf. Concept Mapping (CMC '06)*, vol. 1, pp. 296-303, 2006.
- [50] K. Verbert, J. Jovanović, E. Duval, and D. Gašević, "Ontology-Based Learning Content Repurposing: The ALOCoM Framework," *Int'l J. E-Learning*, vol. 5, no. 1, pp. 67-74, 2006.
- [51] A. Zouaq, R. Nkambou, and C. Frasson, "Bridging the Gap between ITS and eLearning: Towards Learning Knowledge Objects," *Proc. Ninth Int'l Conf. Intelligent Tutoring Systems (ITS '08)*, pp. 448-457, 2008.



Amal Zouaq received the PhD degree in computer science from the University of Montreal in 2008. She is now a researcher in the GDAC Laboratory, University of Quebec at Montreal. Her research interests focus on knowledge representation and extraction, domain ontology generation, competence ontologies, and learning object generation.



Roger Nkambou received the PhD degree in computer science from the University of Montreal in 1996. He is currently a professor of computer science at the University of Quebec at Montreal and the director of the Knowledge Management Research (GDAC) Laboratory (<http://gdac.dinfo.uqam.ca>). His research interests include knowledge representation, intelligent tutoring systems, intelligent software agents, ontology engineering, student modeling, and affective computing. He also serves as a member of the program committee of the most important international conferences in artificial intelligence in education. He is a member of the IEEE.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.