# Evaluating the Generation of Domain Ontologies in the Knowledge Puzzle Project

Amal Zouaq and Roger Nkambou, *IEEE Member*

**Abstract**— One of the goals of the Knowledge Puzzle Project is to automatically generate a domain ontology from plain text documents and use this ontology as the domain model in computer-based education. This paper describes the generation procedure followed by TEXCOMON, the Knowledge Puzzle Ontology Learning Tool, to extract concept maps from texts. It also explains how these concept maps are exported into a domain ontology. Data sources and techniques deployed by TEXCOMON for ontology learning from texts are briefly described herein. Then the paper focuses on evaluating the generated domain ontology and advocates the use of a three-dimensional evaluation: structural, semantic and comparative. Based on a set of metrics, structural evaluations consider ontologies as graphs. Semantic evaluations rely on human expert judgment and finally, comparative evaluations are based on comparisons between the outputs of state-of-the art tools and those of new tools such as TEXCOMON, using the very same set of documents in order to highlight the improvements of new techniques. Comparative evaluations performed in this study use the same corpus to contrast results from TEXCOMON with those of one of the most advanced tools for ontology generation from text. Results generated by such experiments show that TEXCOMON yields superior performance, especially regarding conceptual relation learning.

**Index Terms**— Concept learning, Domain engineering, Knowledge acquisition, Ontology design

## 1 Introduction

Ontologies are the backbone of knowledge representation for the Semantic Web. In the domain of computer-based education, it is believed that ontologies can play a major role in the future of intelligent tutoring systems and eLearning knowledge bases. The educational Semantic Web [2] is an initiative to enrich learning environments with Semantic Web languages and representations. In this context, ontologies can act as a common and reusable knowledge base that training systems can reuse for learning purposes, provided that such systems adhere to the domain knowledge view expressed in the ontology.

In fact, knowledge is never a fixed entity: it evolves with new discoveries and usages. In order to keep ontologies updated with such advances, automatic methods to build them and extend or update them must be set up. The dynamic nature of knowledge implies that manual methods used to build domain ontologies are not scalable: they are time and effort consuming and represent knowledge as a set structure established at the time the ontology was conceived and built.

In order to minimize these drawbacks and avoid the tremendous effort of consistently starting over again, automatic methods for domain ontology building must be adopted. Various domain documents can be used as a source of knowledge. Using domain texts to capture the view of a certain community can help preserve a consensus among community members. Since the ontology "emerges" from texts, it is possible to explain the presence of a particular concept, property, instance or attribute. Hence, ontology learning from texts can help retain somewhat of a semantic validity by providing the means to refer to the original texts.

However, (semi-)automatic methods for ontology learning from text should not be considered as a holy grail that outputs clear and perfect structures. Automatic knowledge extraction techniques can only provide domain ontology skeletons and more complex building steps still require the intervention of human actors.

Another issue within the ontology community pertains to the lack of methodologies to evaluate ontologies, be they built manually or automatically constructed. In fact, a wide adoption of domain ontologies presupposes a means to evaluate the quality, cohesion, domain covering, and richness of these ontologies.

This paper presents TEXCOMON, a knowledge extraction tool produced within the Knowledge Puzzle Project. TEXCOMON provides a solution for the aforementioned issues by generating semi-automatic domain ontologies from texts, and by offering a clear evaluation methodology to analyze ontologies from three perspectives: the structural, semantic and comparative dimensions. The goal of this paper is to present TEXCOMON and the evaluation methodology used to assess the generated domain ontologies. It is organized as follows:

Section 2 briefly presents state-of-the-art elements in the domain of automatic ontology building and evaluation with an emphasis on the evaluation methodologies. Section 3 details the software suite TEXCOMON and explains ontology learning from domain texts. The remainder of the paper focuses on a methodology to evaluate the generated ontology based on structural measures (Section 4) and on comparative measures with one of the most advanced system in domain ontology generation: TEXT-TO-ONTO (Section 4.4). A semantic evaluation is also performed. Finally,

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

Section 5 discusses the results of the evaluation methodology.

## 2 STATE OF THE ART

This section addresses two areas: domain ontology learning and population from text as well as domain ontology evaluation techniques.

### 2.1. Domain Ontology Learning from Texts

Domain ontology learning from text relies on different techniques such as machine-learning and statistical methods [6], [7], [9], linguistic methods [5], [26] or a combination of both [20]. This paper proposes a lexico-syntactic analysis that differs in two ways from existing techniques: First, it is used to extract concept maps from texts and transform them into a domain ontology in a semi-automatic manner, which, to the authors' knowledge, has yet to be attempted. This approach is particularly interesting in cases where sentence structures must be preserved. This is especially important in eLearning, in order to index particular portions of learning objects by way of specific concepts and relationships. Second, aside from integrating well-known linguistic patterns such as [12], [22], the approach proposes a set of domain-independent patterns relying on dependency grammar. Although dependencies have been used to extract information from text [17], this work differs from the existing techniques by the proposed patterns and the methods used to transform instantiated patterns into semantic structures.

Several ontology learning approaches and systems have been proposed over the last decade. Some of them are autonomous ontology learning systems, while others consist of support tools to build ontologies. Two interesting reviews of ontology learning from text are found in [8], [25].

In practical terms, as defined by Shamsfard and Barforoush [28], "an ontology may be defined as O= (C, R, A, Top), in which C represents a non-empty set of concepts (including relation concepts and Top), R the set of assertions in which two or more concepts are related to one another, A the set of axioms and Top the highest-level concept in the hierarchy. R, itself, includes two subsets: H and N: H depicts the set of assertions for which relations are taxonomic and N denotes those which are non-taxonomic."

Thus, ontology learning from texts aims to discover: domain terms, concepts, concept attributes, taxonomic relationships, non-taxonomic relationships, axioms and rules [31].

Figure 1 summarizes what Cimiano et al. call the Ontology Learning Layer Cake [7].

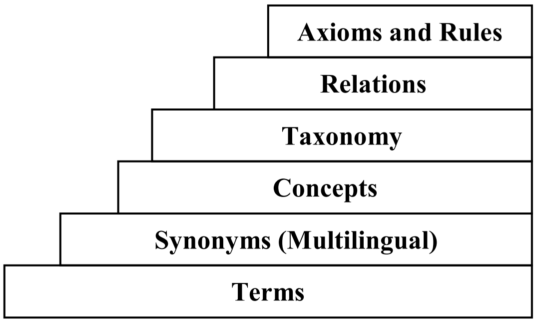



Fig. 1. The Ontology Learning Layer Cake [7].

### 2.2 Evaluating Domain Ontologies

Another complex issue that must be addressed is the evaluation of domain ontologies. People who construct ontologies need tools and methods to evaluate their work and to possibly guide the construction process and refinement steps. Automated or semi-automated ontology learning techniques also require effective evaluation measures, which can be used to select the best of many ontologies, to choose values of tunable parameters of the learning algorithm, or to direct the learning process itself.

Ontologies can be assessed by using different approaches [3]:

- Gold Standards: by comparing the ontology with a "gold standard" [19];
- Application-based: by using the ontology with an application and evaluating the results [23];
- Data-driven: by comparing the ontology with a source of data from the domain to be covered [4];
- Assessment by domain experts [18].

Other approaches attempt to detect the structural properties of the ontology, which is considered as a graph [1], [24], [22]. We believe that structural evaluations, such as the one proposed in [1], are essential and that they must be coupled with some application-based measures. The former determines the structural properties of the ontology; the latter helps to decide how useful the ontology is in a given application scenario. We also believe that human evaluations are essential, especially when such techniques are applied to the field of education.

## 3. THE KNOWLEDGE PUZZLE GENERATION APPROACH: TEXCOMON

The Knowledge Puzzle is a multi-faceted research project to build and exploit knowledge bases in the field of education. One of its goals is to promote ontology learning from any text, and particularly from textual learning objects through the TEXCOMON tool. The generated ontology represents the implicit domain knowledge schema contained in the learning objects.

TEXCOMON, whose name consists of a blend for TEXt-COncept Map-Ontology, is used to indicate the process followed in order to extract domain concept maps from textual documents, and to transform these extracted concept maps into an OWL ontology. Note that the term "concept map" refers to a network of domain terms and relationships extracted from texts and that the conversion of such concept maps into an OWL ontology requires that important concepts and relationships be identified within the concept maps. The importance of concepts will be more precisely defined in subsequent sections.

There are numerous reasons why such a process would be of interest:

Firstly, it allows the creation of concept maps, which have proven their value as knowledge representations and as a way to provide meaningful and constructivist learning [21];

Secondly, it allows for the creation of a formal bridge between concept maps and OWL ontologies, which can be useful [12] since domain experts can model concept maps more easily than ontologies;

Thirdly, it establishes a reusable domain-independent methodology to generate domain ontologies.

A corpus of 36 documents containing approximately 30,000 words was derived from manuals about the SCORM standard [27]. Such a corpus is used for the examples and evaluations reported in the remainder of this paper.

Overall, the ontology engineering process with TEXCOMON works as follows (Figure 2): textual domain documents (textual learning objects and other documents) are used as inputs and an index structure is created by decomposing the document into paragraphs and sentences. Key sentences are then extracted using a machine learning algorithm. These key sentences are then parsed through a statistical natural language processing parser. The parser outputs typed dependency networks. Then the networks are mined in order to identify lexico-syntactic patterns which transform the grammatical representations into semantic ones. The semantic representations are then used to create concept maps. Finally the concept maps are exported as an OWL ontology. The following sections discuss the TEXCOMON process in detail.

### 3.1. Extracting Key Sentences

Paragraphs and sentences are obtained from each document through IBM UIMA-based Java annotators [30]. Key sentences are extracted by running a key sentence extractor that collects sentences which include certain keywords. These keywords are mined through a keyword detection algorithm [11]. Key sentence detection helps reduce the size of the corpus to be analyzed by a linguistic parser. It also helps focus on statistically significant words and their relationships with other words or concepts.

Each sentence is then parsed through the Stanford Parser [16], which outputs a typed dependency network [10], called a grammatical concept map.

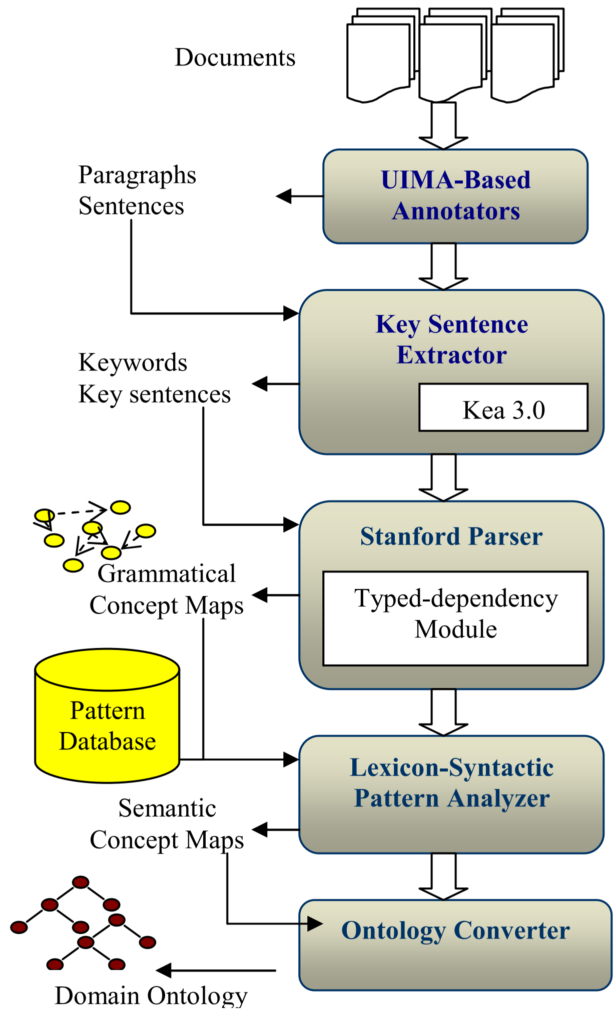



Fig. 2. TEXCOMON Process and Tools

### 3.2. Modeling Patterns through Dependencies

Linguistic analyses can be based on constituency or dependency grammars. Since dependency links are intuitively suitable for semantic interpretations, a dependency representation is selected. Moreover, dependency paths have been used in several models to extract information [17] such as question-answering, paraphrasing, etc., and have shown their validity as knowledge extraction templates.

Since the TEXCOMON objective is to remain domain-independent, a syntax-guided method is proposed to model lexico-syntactic patterns into typed dependencies sub-trees. Each pattern is organized as a tree around a root term, T, which represents a variable that inputs and outputs specific grammatical links. Each node in the pattern represents a variable. During the analysis process, these patterns are sought in the text and instantiated with data whenever an occurrence is found.

A manual analysis of the typed dependencies in the corpus generated the modeling of close to 22 lexico-syntactic patterns, which are organized into terminological and relational patterns and stored in a database.

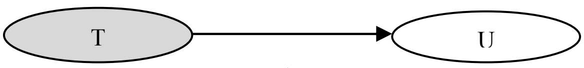
### 3.3. Identifying Terminological Patterns

Domain terms are identified by detecting a set of particular typed dependencies. Some of these typed dependencies directly indicate a domain term (see Table 1).

Table 1. Terminological patterns

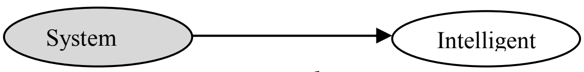
| Pattern (input links) |
|---|
| subj - subject<br>nsubj - nominal subject<br>nsubjpass - passive nominal subject<br>csubj - clausal subject |
| obj - object<br>dobj - direct object<br>iobj - indirect object<br>pobj - object of preposition |
| agent – agent |
| abbrev - abbreviation modifier |
| sdep - semantic dependent |
| appos – appositive |

Other terminological patterns rely on output links from a source term T and need a small structural transformation, by composing a new term from T and its destination. These links are the following: *amod - adjectival modifier* and *nn - noun compound modifier*, as shown below.




*amod* or *nn*

Moreover, the aggregation operation involving an "*amod*" link yields the creation of a taxonomical link between the source term T and the newly composed term U T. For instance:

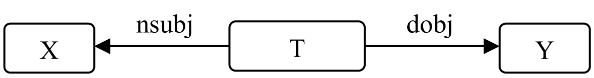



*amod*

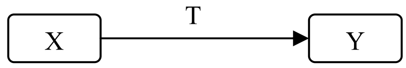
The transformation results in a new composite term, i.e., "Intelligent System" and a taxonomic link: is-a (Intelligent System, System).

### 3.4. Identifying Relational Patterns

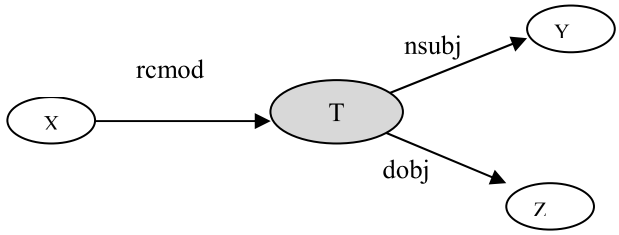
Relational and lexico-syntactic patterns, coupled with Java methods, transform a grammatical structure into a "semantic" one. As with previous terminological patterns, they also rely on detecting sub-graphs in typed dependency networks. An example of a well-known pattern is the *subject-verb-object* pattern, schematized below:




The transformation results into the following triple:




Another pattern is shown below:




In this example, the term T has one input link (*rcmod: relative clause modifier*) and two output links, namely *nominal subject* and *direct object*. This pattern includes an additional constraint: the *nsubj* link must point towards a relative clause modifier (that, which, etc.). The transformation process creates a new relationship labeled as T between the source of the rcmod link (X) and the destination of the dobj link (Z). An occurrence of this pattern can be found in the following sentence: "*The prescription specifies the activities that use the content objects in the package*". In this sentence, the typed dependency network is:

nsubj(specifies-3, prescription-2)
dobj(specifies-3, activities-5)
nsubj(use-7, that-6)
rcmod(activities-5, use-7)
dobj(use-7, objects-10)
nn(objects-10, content-9)
prep_in(objects-10, package-13)

The pattern analyzer creates a semantic relationship "activities – use – content objects" from the above pattern. It also finds an instance of the *subject-verb-object* pattern described above and creates the relationship "prescription – specifies – activities".

The whole process results in a set of semantic relationships constituted by a source term, a destination term and a label. Each of these semantic relationships is stored in a property "Relation", which is linked to its source concept. Semantic relationships contain a pointer towards their originating sentences. This allows for effective indexing by sentence, paragraph and entire document.

### 3.5. Example of a Semantic Analysis

Figure 3 shows a complete example of a transformation process based on lexico-syntactic patterns. The parsed sentence ("An asset is a content object that will not use the SCORM API, but that can still be used for an activity") is decomposed into a set of typed dependencies.

**Step 1:** Removing determiners
nsubj(object-6, asset-2)
cop(object-6, is-3)

| |
|---|
| nn(object-6, content-5)<br>nsubj(use-10, that-7)<br>aux(use-10, will-8)<br>neg(use-10, not-9)<br>dep(object-6, use-10)<br>nn(API-13, SCORM-12)<br>dobj(use-10, API-13)<br>nsubjpass(used-19, that-15)<br>aux(used-19, can-16)<br>auxpass(used-19, be-18)<br>conj_but(use-10, used-19)<br>prep_for(used-19, activity-22) |
| **Step 2 :** Aggregation of complex terms and verbal relationships<br>nsubj(content-5 object-6, asset-2)<br>cop(content-5 object-6, is-3)<br>nsubj(will-8 not-9 use-10, that-7)<br>dep(content-5 object-6, will-8 not-9 use-10)<br>dobj(will-8 not-9 use-10, SCORM-12 API-13)<br>nsubjpass(can-16 be-18 used-19, that-15)<br>conj_but(will-8 not-9 use-10, can-16 be-18 used-19)<br>prep_for(can-16 be-18 used-19, activity-22) |
| **Step 3** : Detection of lexicon-syntactic patterns<br>is-3 (asset-2, content-5 object-6)<br>will-8 not-9 use-10 (asset-2, SCORM-12 API-13)<br>can-16 be-18 used-19 prep_for (asset-2, activity-22) |
| **Step 4** : Number and prefix (prep_, conj_) removal<br>Is (asset, content object)<br>will not use (asset, SCORM API)<br>can be used for (asset, activity) |

Fig. 3: From grammatical to semantic concept maps: the transformation process.

As shown in Figure 3, the first step involves removing non-content words (mostly determiners). The second step pertains to aggregating certain terms in order to generate more complex nouns (composite nouns) or verbs (relationships). Each occurrence of a given expression is replaced by the aggregated terms in all grammatical relationships. For instance, note that "object-6" is replaced by "content-5 object-6" and "use-10" by "will-8 not-9 use-10".

The third step executes the analysis of structures composed of more than one grammatical relationship. For instance, the relationships *nsubj(content-5 object-6, asset-2)* and *cop(content-5 object-6, is-3)* trigger a pattern that creates a new semantic relationship: *is(asset, content object)*.

Another pattern involves a dependency relationship (*dep*):

*dep(content-5 object-6, will-8 not-9 use-10)*
*dobj(will-8 not-9 use-10, SCORM-12 API-13)*

Based on these grammatical links, we can easily deduce: *will-8 not-9 use-10(content-5 object-6, SCORM-12 API-13)*. However, what really matters is that an Asset (not a Content Object) will not use SCORM API. Hence, the pattern retrieves the implicit subject of the dependency relationship. This relationship "*nsubj (content-5 object-6, asset-2)*" enables replacing *content object* by *asset* in the previous relationships, thus resulting in the relationship *will not use (Asset, SCORM API)*.

The semantic analyzer continues searching applicable patterns in the grammatical concept maps until it no longer recognizes any pattern.

### 3.6. Creating Semantic Concept Maps

Aggregating the different relationships of a particular concept makes it possible to create a semantic concept map for this element. This concept map is extracted from various sentences found in numerous documents. Figure 4 shows an example of such a semantic concept map.

As shown in Figure 4, concept maps model relationships in triples but also through **paths of information**. For instance, in the sentence "*Metadata allows for search within repositories*", TEXCOMON is not only able to extract the relationship *allows for (metadata, search)* but also the relationship *within (search, repositories)*, thus adding new knowledge about the primary relationship *allows for*. However, converting concept maps into OWL ontology does not permit these kinds of paths. This raises an interest for both structures:

The domain ontology represents domain concepts and not only domain terms. Thus indentifying significant concepts in the concept maps enables a higher-level indexing of the domain texts. This is performed with metrics from graph theory.

Concept maps represent an additional domain terminology layer: they enrich the domain ontology; they provide users with information paths and allow them to refer back to the original documents.

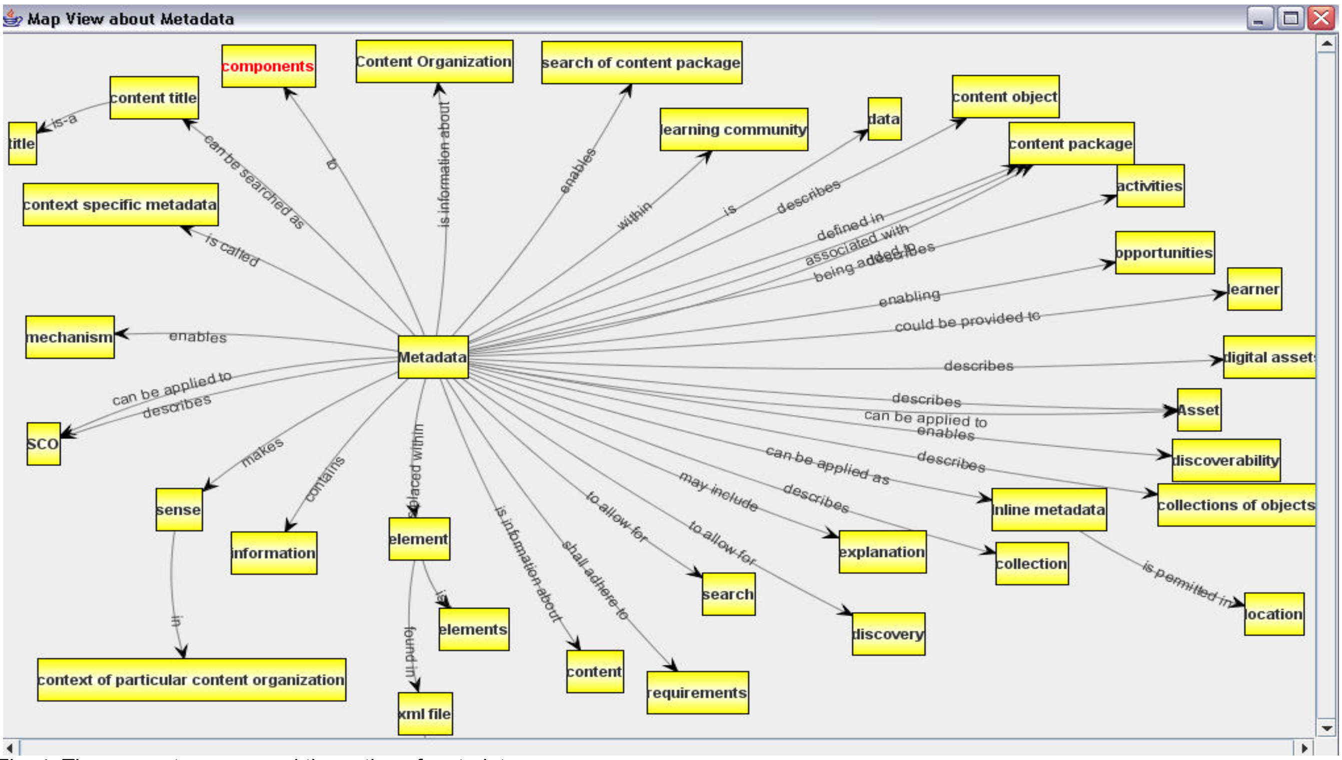



Fig. 4. The concept map around the notion of metadata.

The following section describes the conversion of concept maps into a domain ontology.

### 3.7. Converting Concept Maps into a Domain Ontology

Domain concept maps act as skeletons on which domain ontologies are built. This process implies determining classes, relationships, attributes and instances in the concept maps.

#### 3.7.1. Defining Classes

Extracting ontological classes (concepts) from concept maps is performed by detecting the presence of high density components, which indicate the importance of a given concept. In the Knowledge Puzzle, a term is considered a concept when it is linked to other domain terms through a number of semantic relationships. This number can be parameterized according to the corpus size and the human experts' goals.

A single concept in a text can be expressed in different ways. The Knowledge Puzzle can recognize the base form of a concept through stemming. TEXCOMON uses a Java version of the Porter Stemmer [34] 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 recognizing the plural forms of nouns and certain conjugated verbs. Another way of expressing concepts is through acronyms (e.g., ''SCO'' stands for ''Sharable Content Object''). Although the Stanford University Parser outputs acronym links as typed dependencies, this feature is not always reliable. Hence, the TEXCOMON implements an algorithm to identify correct acronyms, which are stored as terms associated with the current concept.

Most extracted classes belong to primitive classes. However, some defined classes can also be detected through the "abbrev" link. Each concept and its acronym are defined 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 current time, the Knowledge Puzzle lacks the ability of handling anaphors and cannot process antecedents such as ''reference model'' and ''the model'' in the following text: "SCORM is a reference model […]. The model …"

#### 3.7.2. Defining Relationships

Basically, all verbal relationships between pairs of classes are considered as potential ontological relationships. The relationships generated include simple object properties such as:

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

An object property can also take the shape of a blend of classes in its range or domain, as shown below:

```
<owl:ObjectProperty rdf:ID="describes">
```

```
<rdfs:range>
 <owl:Class>
 <owl:unionOf rdf:parseType="Collection">
  <owl:Class rdf:about="#content_objects"/>
      <owl:Class rdf:about="#asset" />
      <owl:Class rdf:about="#Activities" />
      <owl:Class rdf:about="#SCO" />
   </owl:unionOf>
     </owl:Class>
   </rdfs:range>
  <rdfs:domain rdf:resource="#metadata" />
</owl:ObjectProperty>
```

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).

#### 3.7.3. *Defining subclasses, instances and attributes*

Extracting instances enables finding objects which are instances of a particular concept. Hearst [13] first brought up linguistic patterns to identify hyponyms ("is a kind of"). For instance, the pattern "NP1 such as NP2, NP3 and NP4" expresses a hyponymy relationship.

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

•If NP2, NP3, or NP4 are also classes, they are considered sub-classes of NP1.

•Otherwise, if NP2, NP3 and NP4 are not considered classes, they are stored as instances of NP1.

•Finally, if NP1 is not a class as previously defined, then the relationships are left as "is a kind of" between these terms and the human evaluator is free to assign it to a sub-class, an instance or something else.

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 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> a modifier. A sample text that matches this pattern would be: … 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], [33]. If <attr> is a concept, the attribute is considered an OWL Object Property; otherwise it is created as a Data Type Property.

## 4 EVALUATING DOMAIN ONTOLOGIES IN THE KNOWLEDGE PUZZLE

An increased usage of domain ontologies requires a well-established method to evaluate them. This section addresses the evaluation of the domain ontology generated through TEXCOMON based on a number of internal structural measures. The purpose of the evaluation is to assess how well the generated ontology performs, given certain measures.

### 4.1 The Evaluation Method

As indicated below, the evaluation method consists of three kinds of evaluations: structural, semantic and comparative.

The structural evaluation aims at detecting the structural characteristics of the generated domain ontology. Based on different measures, such characteristics can help ontology designers decide what available ontology best suits their ontological needs.

The semantic evaluation involves domain experts who judge the quality of the ontology, or at least the plausibility of its concepts and relationships.

Finally, since generating domain ontologies is far from being perfect in terms of processes and results, one of the most interesting indicators of advancement in the field may consist of testing the available ontology learning tools by comparing the results generated with the very same corpuses. Such a comparative evaluation is designed to offer a basis for new researchers in the field. For the purpose of this study, TEXCOMON was compared to TEXT-TO-ONTO, one of the most advanced tools in the domain of ontology generation. One of the advantages of its approach lies in its availability as an open-source project that can be easily downloaded and tested [29]. Although the methodology used in TEXCOMON differs substantially from the one in TEXT-TO-ONTO, both yield results that can be compared.

### 4.2 Experiment Description

The objective of this experiment is to assess whether or not the generated ontologies represent a given domain as described by keywords previously selected by domain experts. The criteria chosen to measure how these keywords represent the ontology are as follows:

1. The sought terms exist as classes in the ontology;
2. The corresponding classes:

- appear in an adequate structural proximity to one another;
- are described in a rich manner;
- are linked through many relationships;
- appear as central elements in the ontology.

Table 2 shows the keywords chosen as representative concepts for the SCORM standard, an eLearning standard selected as the domain of interest.

Table 2. The set of domain representative sought terms

| Key Search Terms |
|---|
| Asset |
| SCO |
| SCORM Content Model |

| |
|---|
| SCORM |
| LMS |
| Runtime Environment |
| Metadata |
| SCORM Content Packaging |
| Activity |
| Content Organization |
| API |
| PIF |

As mentioned above, with the TEXCOMON approach, experts must assign a value to a parameter that represents the out-degree of a concept. During the experiment, four domain ontologies were generated from the same corpus. These ontologies correspond to different values of a given parameter I. In other words, I=2, which outputs the ontology KP-2; I=4, which outputs the ontology KP-4; etc.

The second step of the experiment consists of performing the same kinds of measures on the ontology generated by TEXT-TO-ONTO and comparing the results in terms of concept existence, richness and interconnection levels.

Seven sub-corpuses are derived from the 36 documents used to assess the evolution of the different metrics when new domain documents are added to the previous corpus in an incremental manner (Table 3). For example, Corpus 2 contains Corpus 1, to which four new files were added.

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 |

## 4.3 Structural Evaluation

The structural evaluation approach is based on a set of metrics defined by [1]. Initially, these metrics were developed to rank ontologies and sort them for retrieval purposes, much like Google and its Page Rank algorithm. Given a set of search terms, Alani and Brewster [1] attempted to find the best ontology to represent these terms.

This first vision was slightly modified by considering an initial set of key search terms as being representative of a domain, in an attempt to identify if the generated ontology includes these terms as classes and to assess how many of these terms are interconnected and richly described by attributes.

The structural metrics are the Class Match Measure (CMM), the Density Measure (DEM), the Betweenness Measure (BEM) and finally the Semantic Similarity Measure (SSM). A total score is then computed from all these measures. This score can be used to rank the ontology with respect to the given search terms. The values of all metrics and final scores are set between 0 and 1.

The functions (ONTO-EVALUATOR library) were implemented to perform different metrics computations based on the exact formulas described in [1]. Jung's Betweenness algorithm [14] was also used to directly calculate the betweens measure.

### *4.3.1 Class Match Measure (CMM)*

The Class Match Measure (CMM) evaluates the coverage of an ontology for the given sought terms.

Given the input sought terms, the ONTO-EVALUATOR searches the classes in the ontology, to determine if the sought terms correspond exactly to ontological classes (exact match) or if they are included in the label of one or many classes (partial match).

Figure 5 shows the evolution of the CMM values for the different corpuses by taking into account partial and exact matches.

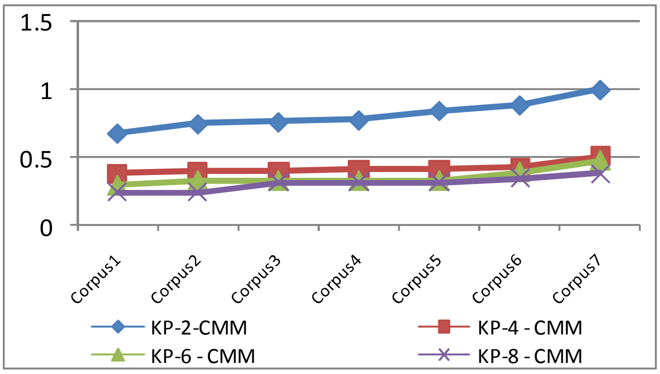



Fig. 5: CMM Evolution across corpuses and thresholds

CMM tends to improve as the threshold decreases in the same corpus. In Figure 5, KP-2 and KP-4 have a higher CMM value. This shows that many concepts which contain the sought terms (partial or total match) are deleted when the threshold increases, thus eliminating important concepts, as defined by the domain expert, that should have been retained otherwise.

An interesting finding: when taking into account only exact matches, that is classes whose labels are identical to the sought term, different graphs are obtained (See Figure 6 below).

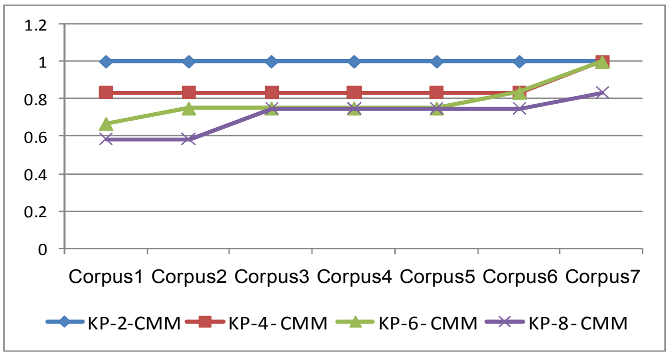



Fig. 6 : CMM resulting from exact match

With Corpus no 7, KP-2, KP-4 and KP-6 achieve identical results. However, KP-8 offers a poorer performance. This indicates that the sought terms, considered as key domain terms, have up to seven relationships with other domain terms.

Considering exact matches and/or partial matches can also affect other metrics. In fact, most of the results are divided by the number of matched classes. When the impact is clearly identified, it is highlighted in the following specific metrics.

### 4.3.2 Density Measure (DEM)

The density measure expresses the degree of detail or the richness of the attributes of a given concept. The underlying assumption is that an adequate representation of a concept must provide sufficient details about this concept. The density measure includes the number of subclasses, of inner attributes, of siblings and of relationships with other concepts.

Figure 7 shows DEM evolution across corpuses and thresholds.

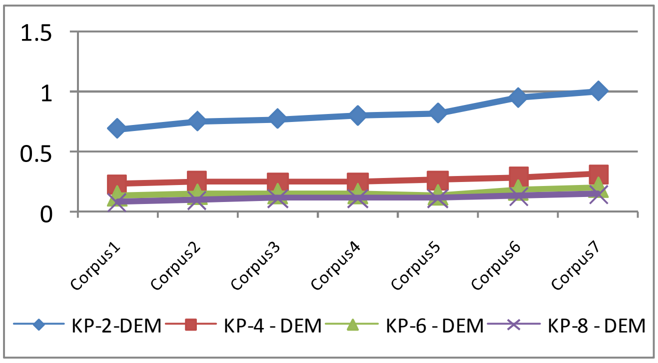



Fig. 7: DEM Evolution across corpuses and thresholds

The DEM tends to increase proportionally to the number of concepts. These variations result from the richness of information in the new corpus. For example, Corpuses 6 and 7 probably add many new relationships, explaining the significant increase, especially when threshold = 2.

### 4.3.3 Betweenness Measure (BEM)

The BEM calculates the betweenness value of each search term in the generated ontologies. Betweenness indicates the extent to which a concept lies on the paths between others. The underlying assumption is that the centrality of a class in an ontology is important. A high betweenness value shows the centrality of this class. As in ActiveRank [1], Onto-EVALUATOR uses the BEM provided by JUNG [14]. This algorithm calculates the number of shortest paths that pass through each concept in the ontology, considered as a graph. A higher betweenness value is assigned to concepts that occur on many ''shortest paths'' between other concepts.

A reasonable number of relationships must be retained in order to have an significant BEM. Figure 8 suggests that again, thresholds 2 and 4 seem to be the best options available.

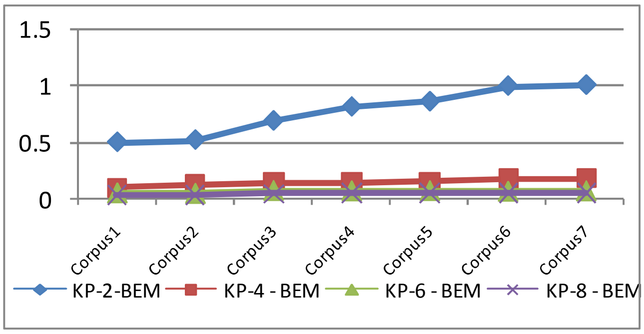



Fig. 8 : BEM Evolution across corpuses and thresholds

### 4.3.4 Semantic Similarity Measure (SSM)

The last measure, the Semantic Similarity Measure (SSM), computes the proximity of the classes that match the sought terms in the ontology. As stated by Alani and Brewster [1], if the sought terms are representative of the domain, then the corresponding domain ontology should link them through relationships (taxonomic or object properties). Failing to do so may indicate a lack of cohesion to represent this domain knowledge.

The SSM is based on the shortest path that connects a pair of concepts. As shown in Figure 9, correlations exist between the text volume and the SSM. The SSM never decreases, regardless of the threshold value. In general, a high threshold value yields a poorer performance of the SSM value. However, with larger corpuses, high thresholds become more appropriate.

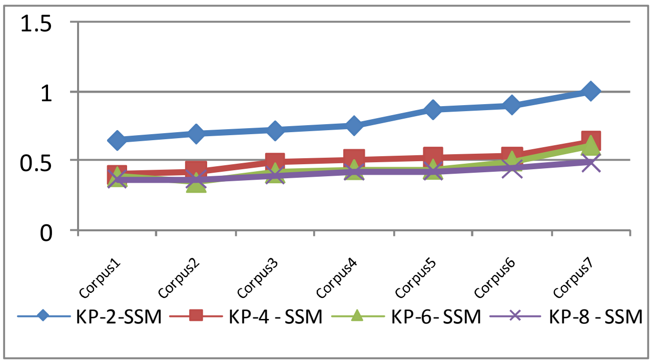



Fig. 9: SSM Evolution across corpuses and thresholds

As previously stated, considering only identical matches has a significant impact on this metric (see Figure 10).

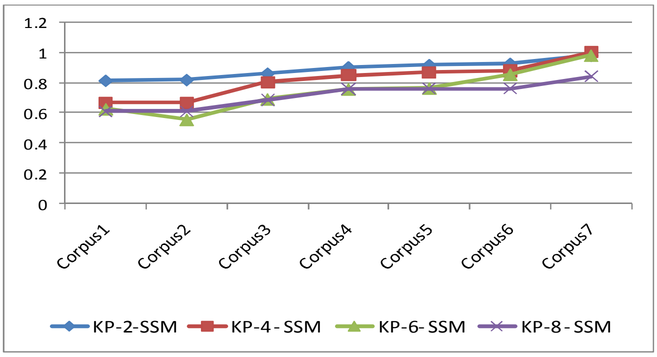



Fig. 10. SSM evolution with exact match

In that case, the exact match leads to very similar results for KP-2, KP-4 and KP-6, especially with the most extensive corpus (no 7) where identical results are obtained. This is not the case if partial and exact matches of input terms are adopted (Note the differences between Figures 9 and 10).

Finally, an overall score can be computed, based on these four metrics. The weights assigned to each of the metrics can be either the same or different. This overall score will be explained in the following section.

### 4.4 Comparative Evaluation

The comparative analysis involves the generation of an ontology with TEXT-TO-ONTO using the same corpuses. Unlike TEXCOMON, TEXT-TO-ONTO has no parameters related to the out-degree of concepts. However, other parameters can be taken into account, more particularly the support given to the association rules generated by the algorithm.

To clarify this notion of support, it is important to first define association rule learning. This type of learning can detect the items (in this case: terms), which co-occur frequently and extract rules that connect such items. The support of an association rule is the percentage of groups (in this case: documents) that contain all the items of the rule.

Two ontologies were generated with TEXT-TO-ONTO (TTO-1, TTO-2) using a total of seven corpuses. Two supports were considered: one of 0, indicating that any processed rule is considered valid (i.e., TTO-1) and the other of 0.1 (i.e., TTO-2). In other words, with TEXT-TO-ONTO, each corpus enables the generation of two different ontologies.

This experiment shows that even a 0.1 support value discards all association rules generated by TEXT-TO-ONTO. In fact, TTO-2 reveals an important disparity of results compared to TTO-1, which contains many meaningless properties that increase the value of certain structural metrics.

To generate a domain ontology using TEXT-TO-ONTO, each of the seven corpuses are used with the KAON Workbench [15]. For each corpus, the following functions are performed:

- Term extractions;
- Instance extractions;
- Association rule extractions with a minimum support of 0 and 0.1. The emerging associations were added to the ontology as properties;
- Relation learning;
- Taxonomy learning using Taxo Builder. A combination-based approach is exploited using Hearst patterns and heuristics. The FCA-based (Formal Concept Analysis) approach is not used because firstly, there are no bases on which to compare TEXCOMON and secondly, the formal concept analysis results failed to convince the authors.
- Neither the Pruner nor the OntoEnricher were used.

In order to ensure the validity of the experiment, certain default parameters were kept. The overall scores generated by the TEXT-TO-ONTO ontologies are obtained with the same metrics as presented above. Not all metrics results are shown here due to a lack of space. The generation process followed by TEXCOMON is identical to the one explained in Section 3.

#### 4.4.1 Overall Score

The total score of an ontology can be calculated once the four measures are applied to the entire set of generated ontologies. The global score is calculated by summing up all the measured values and taking into account the weight of each measure. Varying weights can be helpful to determine the relative importance of each measure for ranking purposes.

By assigning identical weights to all metrics (Table 4, Figure 11), TEXCOMON clearly outperforms TEXT-TO-ONTO in all corpuses. KP-8 is the only ontology that yields a lower score as compared to those of TEXT-TO-ONTO.

Table 4. Ontology Overall Scores and Ranks on the Largest Corpus (no 7) with identical weight (0.25) for all metrics.

| Ontology | Score | Rank |
|---|---|---|
| KP2 | 1 | 1 |
| KP4 | 0.41 | 2 |
| KP6 | 0.34 | 4 |
| KP8 | 0.26 | 6 |
| TTO-1 | 0.38 | 3 |
| TTO-2 | 0.29 | 5 |

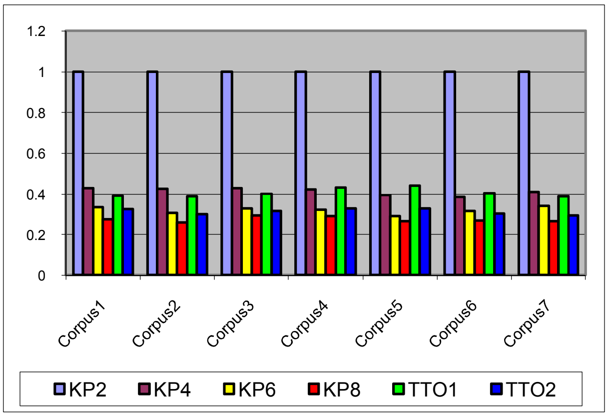



Fig. 11. Overall Score for all corpuses – identical weight for all metrics (i.e. 0.25)

It is also clear from Table 5, that TEXCOMON ontologies perform better than TTO-2 and TTO-1 for KP-2, KP-4 and KP-6 using weights of 0.2, 0.2, 0.2 and 0.4 for CMM, DEM, BEM and SSM, respectively, on the largest corpus.

Table 5. Ontology Overall Scores and Ranks on the biggest Corpus (no 7) with different weights (0.2, 0.2, 0.2, 0.4) for CMM, DEM, BEM and SSM, respectively.

| Ontology | Score | Rank |
|---|---|---|
| KP2 | 1 | 1 |
| KP4 | 0.46 | 2 |

| | | |
|---|---|---|
| KP6 | 0.39 | 3 |
| KP8 | 0.31 | 5 |
| TTO-1 | 0.34 | 4 |
| TTO-2 | 0.24 | 6 |

Moreover, when considering scores resulting from a CMM with identical input terms, the following results are obtained for Corpus 7 with a weight distribution of 0.5, 0, 0, 0.5 (Table 6).

Table 6. Ontology Overall Scores and Ranks on the largest corpus (no 7) with different weights (0.5, 0, 0, 0.5) for CMM, DEM, BEM and SSM, respectively.

| Ontology | Score | Rank |
|---|---|---|
| KP2 | 0.99 | 2 |
| KP4 | 1 | 1 |
| KP6 | 0.99 | 3 |
| KP8 | 0.84 | 4 |
| TTO-1 | 0.62 | 5 |
| TTO-2 | 0.46 | 6 |

For the first time, KP-4 has a better overall score than KP-2. This means that if CMM and SSM are the two most important metrics for a given designer, KP-4 would be the most effective ontology.

One last example considers the score generated by the CMM only. In such a situation, KP-2, KP-4 and KP-6 obtain identical scores and ranks (Table 7).

Table 7. Ontology Overall Scores and Ranks on the largest corpus (no 7) with different weights (1, 0, 0, 0) for CMM, DEM, BEM and SSM respectively.

| Ontology | Score | Rank |
|---|---|---|
| KP2 | 1 | 1 |
| KP4 | 1 | 1 |
| KP6 | 1 | 1 |
| KP8 | 0.83 | 3 |
| TTO-1 | 0.92 | 2 |
| TTO-2 | 0.92 | 2 |

After this macro-level evaluation, we were interested by a micro-level analysis.

### 4.4.2 *Other Comparative Results*

First, certain statistics are compiled of the concepts, relationships and properties generated by TEXCOMON ontologies on the richest corpus (no 7). Then, the same operation is conducted on ontologies generated by TEXT-TO-ONTO and the differences generated by the various ontologies are analyzed (presence or absence of specific properties, concepts, relationships and plausibility of these elements).

Generally speaking, such comparisons clearly show that ontologies generated by TEXCOMON are more interesting, particularly concerning conceptual relationships and when compared to Ontology TTO-2 (support = 0.1). Table 8 compares the results of TEXCOMON and TEXT-TO-ONTO in terms of number of concepts and relationships, both taxonomic and non taxonomic.

Table 8. Certain statistics related to the extracted items

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

This table shows a decrease in the number of concepts and relationships with TEXCOMON. This logical outcome is consistent with the threshold increase. TEXCOMON results can be parameterized which is not the case for TEXT-TO-ONTO where the number of classes in TTO-1 and TTO-2 remains stable. The 33 relationships that appear in "Non taxonomic links" in TTO-2 are the only relationships extracted with a label by TEXT-TO-ONTO. Such a result is disappointing, given the labeled conceptual relations obtained with TEXCOMON.

Another interesting result is found in the difference between the number of non taxonomic links in TTO-2 (33) and TTO-1 (5683). Such a drastic decline is due to the 0.1 support. This indicates that TTO-1 has relations that correspond to association rules with a support that is inferior to 0.1. These relations contribute to the "adequate performance" of TTO-1 in the structural analysis, especially when considering the SSM measure. However, semantically, such relationships are less interesting and should not actually surface in the ontology.

Another way of comparing both systems at a micro level is to take a sought term to watch its corresponding results in the ontologies (KP and TTO). Again, the difference in the results provided for incident edges by TTO-1 and TTO-2 are significant.

Table 9 illustrates these statistics for the terms "SCO" and "asset". Other key domain terms yielded similar results.

Table 9. Statistics pertaining to the domain terms ''SCO'' and ''asset''

| **Term** | **Type** | **KP-2** | **KP-4** | **KP-6** | **KP-8** | **TTO-1** | **TTO-2** |
|---|---|---|---|---|---|---|---|
| SCO | Super classes | 3 | 2 | 4 | 4 | 1 | 1 |
| | Sub classes | 2 | 2 | 2 | 2 | 2 | 2 |
| | Siblings | 6 | 0 | 0 | 0 | 2 | 2 |
| | Incident edges | 18 | 12 | 10 | 7 | 118 | 0 |
| Asset | Super classes | 3 | 2 | 2 | 2 | 1 | 1 |
| | Sub classes | 0 | 0 | 0 | 0 | 0 | 0 |
| | Siblings | 7 | 1 | 0 | 0 | 4 | 4 |
| | Incident edges | 11 | 8 | 7 | 4 | 172 | 0 |

Note that TEXCOMON discovers more super-classes and siblings for these two concepts (especially for KP-2 and KP-4). Also, note the disproportion between the number of properties in TTO-1 and TTO-2 and between TTO-1, TTO2 and KP ontologies. Again, the 172 relations generated in TTO-1 turn into a null value in TTO-2.

Table 10 shows two excerpts of the generated relationships regarding the terms SCO and asset in TEXCOMON.

Table 10. An excerpt of the relationships generated for the terms asset and SCO in TEXCOMON ontologies.

| Terms | Relationships Generated |
|---|---|
| Asset | can_be_used_for – activity<br>will_not_be_included_as - physical_files<br>is_taken_away_from - learner<br>is_basic_building_block_of - training_resources<br>does_not_communicate_to - LMS |
| SCO | is_collection_of - Asset<br>are - responsibilities<br>is_tracked_by - LMS<br>must_behave_within - runtime_environment<br>may_initiate - communication<br>must_able_to_consistently_find - API_Instance<br>can_be_described_with - Metadata<br>to_find - API_Instance<br>is_intended_to_be_delivered_in - LMS<br>may_communicate_to - LMS<br>terminates - communication<br>is_required_to_adhere_to - requirements<br>finds - API_Instance |

An excerpt of the relationships generated by TEXT-TO-ONTO was also investigated regarding the term asset (Table 11). The same kind of results was found for the term "SCO" and was not shown below.

Table 11. An excerpt of the incident edges for the concept "asset" in TTO- 1.

| Term | Relationships Generated |
|---|---|
| Asset | defaultProperty1,068 - launch<br>defaultProperty1,971 - train<br>defaultProperty3,494 - lm<br>defaultProperty1,912 - refer<br>defaultProperty690 - creat<br>defaultProperty2,525 - metadata<br>defaultProperty1,631 - learner<br>defaultProperty3,066 - docum<br>defaultProperty472 - experi<br>defaultProperty1,346 - packag |

The problems with TEXT-TO-ONTO are the following: 1) it does not extract relationship labels between concepts in association rule learning and 2) it fails to keep the complete label of a concept, storing only its stem (in the OWL ontology). TEXCOMON extracts both types of labels (complete labels and stems) and it also extract labels for relationships.

### 4.5 Semantic Evaluation

The third component of this analysis, the semantic evaluation, relies on human experts to assess the validity of the ontology. The semantic evaluation is aimed at detecting to what degree, and how well, the generated domain ontologies reflect the domain knowledge. It is believed that such an evaluation can only be performed by domain experts. Two experts were asked to analyze the resulting ontologies and to remove all inappropriate concepts, attributes and relationships. The percentage of pertinent concepts and relationships was then calculated following the assessment by each expert. This operation was performed on the TEXCOMON and the TEXT-TO-ONTO ontologies.

The following table summarizes the evaluation of the TEXCOMON ontologies, using the mean scores for relevancy, as expressed by the two experts. As shown in the table, results are promising. The same procedure is then repeated with TEXT-TO-ONTO ontologies (Table 12).

Table 12. Mean scores for relevant data generated by both solutions in %).

| Ontology | Primitive Classes | Defined Classes | Hierarchical Relationships | 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 |
| TTO-1 | 73.06 | n/a | 47.53 | 0.31 |
| TTO-2 | 73.06 | n/a | 47.53 | 53.03 |

According to the results of the semantic evaluation, TEXCOMON ontologies yield a superior performance. This is even more significant in conceptual relationships learning, one of the strengths of TEXCOMON, as well as one of the most challenging tasks in text mining. The results of the semantic evaluation confirm those of previous structural and comparative analyses. However, it cannot be disregarded that this novel algorithm takes much time to process the entire corpus (a total of approximately 5 hours) while TEXT-TO-ONTO outputs results much more quickly.

## 5 RESULT ANALYSES AND DISCUSSION

From the results of this experiment, it can be seen that TEXCOMON ontologies yield superior performance than those of TEXT-TO-ONTO, especially when compared to TTO-2.

Revealing results are found when varying weights, exact matches and by observing their impact on the overall ontology scores. The following critical questions must be considered when modifying such parameters:

First, and most importantly, is it possible to obtain a more compact ontology that preserves the performance or the score levels of KP-2?

Second, which are the most important metrics according to the domain, the goal and the needs?

If the answer to the first question is affirmative, then a more compact ontology should be favored over one that is less compact, since it includes more richly interconnected concepts while preserving the sought domain terms. For example, in Table 6, KP-4 should be chosen whereas Table 7 indicates that KP-6 is the best ontology: its score is identical to that of KP-2 and KP-4, but it is much more compact than the latter two.

Table 8 depicts comparative data for the output of both ontologies in terms of number of concepts and relationships (taxonomic and non-taxonomic). Note that:

TEXCOMON results can be parameterized. In fact, any given ontology designer may be interested in larger or more condensed ontologies and should be given the opportunity of fine-tuning results.

With TEXCOMON, a threshold increase is proportional to a decreasing number of concepts and relationships.

A revealing aspect appears in the number of non-taxonomic links in TTO-2 (n=33) compared to TTO-1 (n=5683). This tremendous decrease pertains to the support of 0.1 used by TTO-2, meaning that TTO-1 created relationships corresponding to association rules whose support is inferior to 0.1. Although such relationships contributed to the illusion of an improved performance of TTO-1, especially with SSM measures, they are actually meaningless and offer no added value for ontological relationships.

Given that the majority of relations come from detecting association rules, TEXT-TO-ONTO rarely extracts labels for relations between concepts, unlike TEXCOMON. This lack of labels translates into a label as "defaultProperty" and, subsequently, it is rather difficult to assign meaning to this type of relationship.

In the end, what kind of results can be deduced from the above statements? In the case of the structural evaluation, TEXCOMON offers the possibility of calibrating thresholds according to ontology designers' needs and goals. Given a set of search terms which are considered important domain concepts:

Threshold calibration can be performed by taking into account CMM, if the most important feature is the partial or complete match of search terms as ontological concepts.

If the important feature consists of generating richly described concepts with a significant number of attributes and relationships, then the density measure should have a larger weight in the overall evaluation.

If the important feature consists of finding richly interconnected concepts in order to make them central to the ontology, semantic similarity and betweenness should be considered.

Our opinion is that all the measures are important. In general, and if we take into account the overall score, ontologies KP-2 and KP-4 seem satisfactory, given the corpus size. However, one should bear in mind that structural evaluations alone do not suffice and that they can be misleading when used alone. For instance, the structural evaluation awarded a better score to TTO-1 in comparison with TTO-2 while TTO-2 generated much more significant results in terms of semantics. The semantic evaluation confirmed the results of the structural assessment. The results of this analysis point towards the choice of Ontology KP-4, which offers the best relevancy rate for pertinent classes and hierarchical relationships. As far as conceptual relationships are concerned, there are no significant differences between KP-4, KP-6 and KP-8.

In summary, there is no single right way of evaluating ontologies. However, certain lessons can be retained from this experiment:

In the absence of a gold standard evaluation method for a given domain ontology, building a new one is not always possible. In such a case, another ontology evaluation method must be undertaken;

Comparing the generated domain ontology with others generated by state-of-the-art tools can be beneficial to highlight the added value of the new tool or platform. This confirms the interest of the comparative evaluation as proposed in this paper;

Evaluating an ontology from a structural point of view can also be interesting as shown in [1]. Comparing this structural evaluation as we did with other generated ontologies, is meaningful. However, this comparison should be accompanied with stringent semantic evaluations.

## 6 Conclusion

This paper presented domain ontologies generated in the Knowledge Puzzle Project through TEXCOMON, by focusing on ontology evaluations. Built on Alani and Brewster metrics [1], a complete structural evaluation was presented. A comparative evaluation was also proposed with state-of-the-art ontology generation tools and more specifically with TEXT-TO-ONTO [20]. The goal was to compare the ontological outputs in terms of concepts, attributes, hierarchical and non-taxonomic relationships. Overall scores were computed, based on structural metrics, in order to provide a glimpse into how ontologies can be compared, using various parameters. Finally, a semantic evaluation was performed.

When compared with TEXT-TO-ONTO, TEXCOMON produced more interesting results in terms of concepts and relationships. However, this does not mean that there is no room for improvement. In fact, a lot of noise is generated by lexico-syntactic patterns and their associated methods. Further work must be performed in order to enhance these patterns. Efforts must also be invested in order to reduce the overall processing time of documents. It was noticed that Protégé database projects tend to slow down as the number of instances increases. Moreover, it seems that the OWL Java API used in the Knowledge Puzzle Project could also be improved in terms of processing time. Finally, other experiments must be performed to determine how much of the knowledge contained in the documents is actually retrieved by TEXCOMON. For now, solutions to these specific challenges have yet to be found.

There is no single best or preferred approach to ontology evaluations: the choice of a suitable approach must reflect the purpose of the evaluation, the application in which the ontology will be used and certain specific aspects of the ontology under evaluation. In our opinion, future work in this area should focus particularly on automated ontology evaluations, a necessary prerequisite for the fruitful development of automated ontology processing techniques in order to solve a number of problems, such as ontology learning, population, mediation and matching.

## REFERENCES


[1] H. Alani and C. Brewster, "Metrics for Ranking Ontologies," *Proc. of 4th Int. EON Workshop, 15th Int. World Wide Web Conf.*, Edinburgh, 2006.

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

[3] J. Brank, M. Grobelnik, and D. Mladenić, "A survey of ontology evaluation techniques," *Proc. of the 8th Int. multi-conference Information Society IS-2005*, Ljubljana, Slovenia, 2005.

[4] C. Brewster, H. Alani, D. Dasmahapatra, and Y. Wilks, "Data driven ontology evaluation," *Proc. of International Conference on Language Resources and Evaluation (LREC)*, pp. 26–28, 2004.

[5] P. Buitelaar, D. Olejnik, and M. Sintek, "A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis," *Proc. of the 1st European Semantic Web Symposium (ESWS)*, Heraklion, Greece, 2004.

[6] S. Cederberg and D. Widdows, "Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hyponymy Extraction," *Proc. of the Conference on Natural Language Learning (CoNNL)*, Edmonton, Canada, 2003.

[7] P. Cimiano, "Ontology Learning and Population from Text," PhD thesis at the Universität Karlsruhe (TH), 2006.

[8] P. Cimiano, J. Volker, and R. Studer, "Ontologies on Demand? – A Description of the State-of-the-Art, Applications, Challenges and Trends for Ontology Learning from Text," *Information, Wissenschaft und Praxis* 57 (6-7): 315-320. October 2006.

[9] P. Cimiano, A. Hotho and S. Staab, "Comparing Conceptual, Partitional and Agglomerative Clustering for Learning Taxonomies from Text," *Proc. of the European Conference on Artificial Intelligence (ECAI'04)*, pp. 435-439, IOS Press, 2004.

[10] M-C. De Marneffe, B. MacCartney and C.D. Manning, "Generating Typed Dependency Parses from Phrase Structure Parses," *Proc. of the 5th Conference on Language Resources and Evaluation*, pp. 449-454, Genoa, Italy, 2006.

[11] E. Frank, G.W. Paynter, I.H. Witten, C. Gutwin, and C.G. Nevill-Manning, "Domain-specific key phrase extraction," *Proc. of the 16th International Joint Conference on Artificial Intelligence*, pp. 668-673, San Francisco, USA, 1999.

[12] P. Hayes, T. Eskridge, R. Saavedra, T. Reichherzer, M. Mehrotra, and D. Bobrovnikoff, "Collaborative Knowledge Capture in Ontologies," *Proc. of the 3rd International Conference on Knowledge Capture (K-CAP'2005)*, pp. 99-106. ACM press, 2005.

[13] M. Hearst, "Automatic Acquisition of Hyponyms from Large Text Corpora," *Proc. of the Fourteenth International Conference on Computational Linguistics*, pp. 539–545, Nantes, 1992.

[14] http://jung.sourceforge.net/

[15] http://kaon.semanticweb.org/

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

[17] D. Lin and P. Pantel, "Discovery of inference rules for question answering," *Natural Language Engineering,* 7(4):343–360, 2001.

[18] A. Lozano-Tello and A. Gomez-Perez, "Ontometric: A method to choose the appropriate ontology," *J. Datab. Mgmt.*, 15(2):1–18, 2004.

[19] A. Maedche and S. Staab, "Measuring similarity between ontologies," *Proc. of CIKM 2002,* LNAI vol. 2473, pp. 251 – 263, Springer-Verlag, London, UK, 2002.

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

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

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

[23] R. Porzel, and R. Malaka, "A task-based approach for ontology evaluation," *Proc. of ECAI 2004 Workshop Ont. Learning and Population*, pp. 9–16, 2004.

[24] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and application of a metric on semantic nets," IEEE *Transactions on Systems Management and Cybernetics*, 19(1):17–30, 1989.

[25] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," *Journal of Artificial Intelligence Research,* 11:95–130, 1999.

[26] A. Schutz, and P. Buitelaar, "RelExt: A Tool for Relation Extraction in Ontology Extension," *Proc. of the 4th International Semantic Web Conference*, pp. 593-606, 2005.

[27] SCORM (Sharable Content Object Reference Model). Retrieved September 5th, 2007 from http://www.adlnet.gov/scorm/index.aspx

[28] M. Shamsfard, and A. Barforoush, "The state of the art in ontology learning: a framework for comparison," *Knowl. Eng. Rev.* 18 (4): 293-316, 2003.

[29] http://sourceforge.net/projects/texttoonto

[30] UIMA (2007). Retrieved August 28th, 2007 from: http://uima-framework.sourceforge.net/

[31] J. Völker, P. Hitzler, and P. Cimiano, "Acquisition of OWL DL Axioms from Lexical Resources," *Proc. of the 4th European Semantic Web Conference (ESWC'07)*, pp. 670-685, Springer, 2007.


**Amal Zouaq** received her Ph.D. (2008) in Computer Science from the University of Montreal. She is now a researcher at the GDAC laboratory (Knowledge Management Lab, University of Quebec at Montreal (UQAM)). She is also a lecturer in the Computer Science Department at the University of Quebec at Montreal. Her research interests include knowledge representation and extraction, Semantic Web, data mining, ontology engineering and computational linguistics.

**Roger Nkambou** is currently a Professor in Computer Science at the University of Quebec at Montreal, and Director of the GDAC (Knowledge Management Research) Laboratory (http://gdac.dinfo.uqam.ca). He received a Ph.D. (1996) in Computer Science from the University of Montreal. His research interests include knowledge representation, intelligent tutoring systems, intelligent software agents, ontology engineering, student modeling and affective computing. He also serves as member of the program committee of the most important international conferences in Artificial Intelligence and Education.