Towards Open Ontology Learning and Filtering

Amal Zouaq[1, 3], Dragan Gasevic[2, 3], Marek Hatala[2]

[1] Department of Mathematics and Computer Science, Royal Military College of Canada, CP 17000, Succursale Forces, Kingston, ON K7K 7B4 Canada

[2] School of Interactive Arts and Technology, Simon Fraser University Surrey, 13450 102 Ave. Surrey, BC V3T 5X3, Canada

[3] School of Computing and Information Systems, Athabasca University, 1 University Drive, Athabasca, AB T9S 3A3, Canada

amal.zouaq@rmc.ca, dgasevic@sfu.ca, mhatala@sfu.ca

Abstract. Open ontology learning is the process of extracting a domain ontology from a knowledge source in an unsupervised way. Due to its unsupervised nature, it requires filtering mechanisms to rate the importance and correctness of the extracted knowledge. This paper presents OntoCmaps, a domain independent and open ontology learning tool that extracts deep semantic representations from corpora. OntoCmaps generates rich conceptual representations in the form of concept maps and proposes an innovative filtering mechanism based on metrics from graph theory. Our results show that using metrics such as Betweenness, PageRank, Hits and Degree centrality outperforms the results of standard text-based metrics (TF-IDF, Term Frequency) for concept identification. We propose voting schemes based on these metrics that provide a good performance in relationship identification, which again provides better results (in terms of precision and F-measure) than other traditional metrics such as Frequency of co-occurrences. The approach is evaluated against a gold standard and is compared to the ontology learning tool Text2Onto. The OntoCmaps generated ontology is more expressive than Text2Onto ontology especially in conceptual relationships and leads to better results in terms of precision, recall and F-measure.

Keywords: Ontology learning, filtering, metrics, graph theory.

1 Introduction

With the explosion of the amount of electronic data, in domain-dependent corpora and the Web, the ability of creating conceptual models from textual data is a key issue for the current Semantic Web and Artificial Intelligence research. In fact, the Semantic Web relies heavily on domain ontologies as conceptual models, which aim at making machines able to interpret the actual Web content. However, a well-known problem of the Semantic Web is the knowledge acquisition bottleneck that results from the difficulty of manually building domain ontologies and making them evolve to reflect the actual data content. For this reason, there is a need of semi-automatic methods for building domain ontologies that help domain experts to deal with huge amounts of data. There have been many attempts to reduce this bottleneck through ontology learning tools such as Text-To-Onto [Maedche & Volz, 2001], Text2Onto [Cimiano & Volker, 2005], OntoLearn [Navigli & Velardi, 2004] and OntoGen [Fortuna et al., 2004]. However, these tools suffer from a number of shortcomings that hinder their ability to effectively help the domain expert:

1. They generally generate very shallow and lightweight ontologies due to their reliance on shallow natural language processing (NLP) and stochastic methods. While "A little semantics goes a long way" as stated by [Hendler et al., 2003], there are many application domains that require more expressive ontologies [Volker et al., 2008];
2. They are not designed with the user in mind [Hatala et al., 2009]. However, in a semi-automatic process, building effective user-centered interfaces and processes is an essential step towards the success of the tool. In particular, a previous study in our research group [Hatala et al., 2009] showed that users are overwhelmed by the huge number of concepts proposed by the tool without any other guidance.

Moreover, one other weak point of ontology learning approaches is that they do not require particular characteristics about the knowledge source used to extract the ontology. However, the quality of the generated domain ontology will heavily depend on the quality of its source corpora.

In order to deal with the abovementioned issues, there is a need to set up a methodology for more effectively generating a domain ontology in a **semi-automatic**, **open, unsupervised** and **domain-independent** manner.

This methodology should answer the shortcomings highlighted above by carefully choosing the source of knowledge, adopting deep NLP techniques, filtering the extracted knowledge and presenting adequately the results to users. In particular, the methodology should exhibit the following characteristics [Zouaq et al., 2011]:

- It should be unsupervised and should not rely on hand-made semantic resources like frames, ontologies and pre-existing templates, due to the large effort required from domain experts to develop them;
- It should be able to filter the noise produced by the unsupervised extraction. In fact, the "blinder" the approach is, the more likely it is to generate noisy knowledge;
- It should integrate the knowledge coming from different texts [Kim et al., 2009] and recognize various references to the same elements from various sources.

While the majority of the approaches are domain-independent, very few, if any, recur to deep semantic analysis to extract ontological elements. In particular, the issue of filtering the knowledge is generally addressed using traditional metrics from information theory and targets specifically concept extraction. This paper introduces OntoCmaps, an open ontology learning tool, which creates graph structures, called concept maps, from domain corpora. OntoCmaps relies on deeper semantic analysis methods than what is currently proposed by state-of-the-art ontology learning tools. Consequently, it creates graph structures densely connected, thus generating richer conceptual relationships than the state-of-the-art ontology learning tools. To be able to choose the most relevant concepts and relationships extracted from texts, OntoCmaps requires filtering methods to be adapted to the generated structures. Accordingly, after briefly presenting the OntoCmaps extraction process, the paper focuses on the issue of filtering the extracted data with an innovative approach: since the extracted representations are graph-based, we propose the use of graph theory measures to identify the important components in the graphs. Traditionally, filtering in ontology learning tools remains generally dependent on statistical measures such as TF-IDF and C/NC value (for concepts) [Cimiano & Volker, 2005]. As shown in Sect. 5, this paper demonstrates that it is possible to obtain better results than these standard measures using metrics from graph theory and using semantic relatedness measures to filter out important concepts and relationships, which are then promoted as elements of the domain ontology. To our knowledge, this has not been proposed until now. Our assumption is that these metrics may provide some evidence on the relevance of concepts and relationships without recurring to any external structured knowledge source (e.g., taxonomy, ontology, or dictionary), as this is usually done in the state of the art [Soderland & Mandhani, 2007]. The approach is evaluated at the ontology learning level by comparing the ontology generated by OntoCmaps with the ontology of a state-of-the-art ontology learning tool Text2Onto [Cimiano & Volker, 2005] using a gold standard. It is also evaluated at the filtering level, by comparing the metrics from graph theory to traditional metrics from information theory and to a naïve random baseline.

In this context, the objectives of this paper are:

1. To introduce the OntoCmaps ontology learning tool with the emphasis on its deep semantic analysis and filtering components;
2. To propose a filtering method based on metrics and combination of metrics (a voting scheme) to rank concepts and relationships and extract the most relevant ones;
3. To determine empirically the metrics, which are the most likely to give the results with the highest precision;
4. To compare the results of each metric to those of standard weighting schemes (e.g., TF-IDF, Point-wise mutual information, and Frequency of co-occurrences);
5. To assess the results of all the metrics using a human gold standard and a random baseline; and
6. To assess the results by comparing them with a state-of-the-art ontology learning tool, Text2Onto [Cimiano & Volker, 2005] on the same corpora and against the same gold standard.

This paper is organized as follows. After the introduction, Section 2 presents the motivation of our approach and positions our proposal. Section 3 presents the information extraction process using deep NLP. Section 4 focuses on the filtering process, introduces our hypotheses regarding concepts and relationships relevancy, as well as the metrics for ranking concepts and relations based on graph theory. Section 5 talks about our experiments and compares the results with other standard measures as well as with a human ranking of concepts and relationships. Finally, section 6 presents a set of related work and section 7 summarizes the paper and discusses future work.

2 Motivation

2.1 Deep Semantic Analysis Methods

As previously stated, existing ontology learning approaches and tools are mainly based on shallow NLP analysis techniques and stochastic methods. Shallow techniques do not attempt to achieve an exhaustive linguistic analysis and they ignore many details in the input and the linguistic framework [Schafer, 2007][Xu & Krieger, 2003][Crysmann et al., 2002]. They deliver partial, non-exhaustive and sometimes erroneous representations as it will be demonstrated in the examples below. When applied to ontology learning, these techniques result into very shallow ontologies and lead especially to a lack of conceptual relations between the extracted concepts as it will be shown in the evaluation section (Sec. 5) of this paper. In fact, adequately extracting relations requires more than shallow NLP. For instance, consider these two examples extracted from [Schafer, 2007]:

1. Things would be different if Microsoft was located in Georgia
2. The National Institute for Psychobiology in Israel was established in May 1971 as the Israel Center for Psychobiology by Prof. Joel.

In the first example, a shallow NLP component based on regular expressions would extract that *Microsoft was located in Georgia*, while in the second example, it would extract that *Israel was established in May 1971*, both assertions being wrong [Schafer, 2007]. Many similar errors can be made by relying on shallow extraction techniques. On the contrary, not only is deep NLP able to determine correct arguments for relations, but it is also very useful to identify negation scope, quantifiers' expressions, and other linguistic phenomena useful for building complete and accurate semantic representations leading to more expressive ontologies [Zouaq & Nkambou, 2009][Jiang & Tan, 2010]. That said, statistical methods for NLP are the major trend over the last few years and they have contributed to major advances in the field [Bos, 2009]. This is the reason why OntoCmaps relies on a statistical syntactic parser, the Stanford parser [Klein & Manning, 2003], as its module for extracting syntactic representations.

2.2 Filtering based on Graph Metrics

The filtering issue is very important when the adopted approach for knowledge extraction is an open and unsupervised one. In fact, most of the automatic approaches for semantic analysis and knowledge extraction [Lin et al., 2009] [Gordon et al., 2010] generate a lot of noisy data. This fact is worsened when the extraction relies on various analysis stages that can each contribute to errors due to improper syntactic or semantic analysis which might propagate to further stages. Therefore, there is a need of adequate filtering mechanisms that rank the accuracy or probability of the extracted elements. In the ontology learning community, this is usually done using standard measures of information retrieval such TF-IDF [Salton & Buckley, 88] and C/NC value [Frantzi & Ananiadou, 99] or using redundancy [De boer et al., 2007]. However this ranking is generally limited to concepts. In the context of deep NLP, and with the new ability to generate dense conceptual representations in the form of graphs, as we propose here with OntoCmaps, our initial hypothesis was that metrics from graph theory might provide enough evidence to perform adequately this required filtering step and to identify the important elements in the graphs. In fact, these metrics have been widely used in social network analysis and have been applied to ontologies for the purpose of analysis [Alani & Brewster, 2006] [Hoser et al., 2006], but not for filtering purposes to the best of our knowledge. There are however new initiatives [Coursey & Mihalcea, 2009][Xie, 2005][Zouaq, 2008][Ozgur et al., 2008][Navigli & Lapata, 2010] in the text mining community that show that these metrics might be beneficial for extracting important data in network structures, and that measures such as degree, eigenvector, betweenness and closeness centrality might be useful in many automatic extractions and filtering tasks such as important gene-disease associations discovery [Ozgur et al., 2008] and noun phrase prediction [Xie, 2005]. In the same line of research, this paper proposes a set of metrics that are used to identify graph elements' importance for ontology learning.

3 Semantic Analysis in OntoCmaps

OntoCmaps, which is introduced in this paper, is the successor of TEXCOMON [Zouaq, 2008][Zouaq & Nkambou, 2009] and it is based on a domain-independent, unsupervised, open and deep knowledge extraction approach. OntoCmaps relies on three main phases to learn a domain ontology: 1) the **extraction phase** that performs a deep semantic analysis and extracts various chunks of knowledge (domain terms, is-a relationships, and conceptual relationships); 2) the **integration** phase that builds concept maps, which are composed of terms and labeled relationships, and relies on basic disambiguation techniques; and finally 3) the **filtering phase** where various metrics are used to filter out the obtained concept maps. The filtered concept maps are called *ontological maps* and are then validated and exported into a domain ontology.

A concept map about a term *t* can be defined as a source root element *t* linked to various other terms through taxonomical links (i.e., generalization/specialization) and conceptual links. These maps in turn may or may not be interlinked depending if relationships have been identified between their elements. Note that there is a difference between these concept maps which are in fact term maps that originate from the semantic analysis of texts and **ontological maps**, which are obtained once the filtering of concept maps is performed.

In the **extraction phase**, OntoCmaps relies on a grammar of syntactic patterns which creates semantic representations from syntactic dependency relationships. Furthermore, OntoCmaps does not rely on any predefined domain-dependent template to extract the semantic representations. It uses solely two linguistic components to obtain the syntactic inputs: the Stanford Parser along with its dependency module [De Marneffe et al., 2006] and the Stanford POS Tagger [Toutanova et al., 2003]. The Stanford dependency module generates syntactic dependency relations between the related words of a sentence. The POS Tagger creates parts-of-speech for each word in the sentence. Based on these two inputs, OntoCmaps creates a condensed syntactic representation by enriching the dependency elements with their parts-of-speech as we proposed in [Zouaq et al., 2010]. For example, for the sentence: "*SCORM and the IMS SS Specification are application profiles of the IMS Content Packaging Specification and as such they add a couple of restrictions*", the Stanford parser produces the following representations:

1. SCORM/NNP and/CC the/DT IMS/NNP SS/NNP Specification/NNP are/VBP application/NN profiles/NNS of/IN the/DT IMS/NNP Content/NNP Packaging/NNP Specification/NNP and/CC as/RB such/JJ they/PRP add/VBP a/DT couple/NN of/IN restrictions/NNS
2. nsubj(profiles-9, SCORM-1)
   det(Specification-6, the-3)
   nn(Specification-6, IMS-4)
   nn(Specification-6, SS-5)
   conj_and(SCORM-1, Specification-6)
   cop(profiles-9, are-7)
   nn(profiles-9, application-8)
   det(Specification-15, the-11)
   nn(Specification-15, IMS-12)
   nn(Specification-15, Content-13)
   nn(Specification-15, Packaging-14)
   prep_of(profiles-9, Specification-15)
   advmod(such-18, as-17)
   conj_and(Specification-15, such-18)
   nsubj(add-20, they-19)
   dep(profiles-9, add-20)
   det(couple-22, a-21)
   dobj(add-20, couple-22)
   prep_of(couple-22, restrictions-24)

OntoCmaps merges both representations into a single one by adding the parts-of-speech to the dependency relations (e.g., *nsubj(profiles-9/NNS, SCORM-1/NNP)*). This representation is then exploited to detect particular syntactic patterns, which are mapped to semantic representations through rules that apply transformations on the input representations. In the example sentence, OntoCmaps is able to detect the following relations:

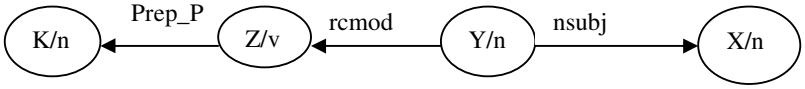
1. Is_a(SCORM, application profiles of the IMS Content Packaging Specification)
2. Is_a(IMS SS Specification, application profiles of the IMS Content Packaging Specification)

3. Is_a(couple of restrictions, couple)
4. Is_a(profiles of the IMS Content Packaging Specification, profile)

The patterns are divided into conceptual patterns and hierarchical patterns. Conceptual patterns identify the main structures of the language. Those structures can be directly mapped to conceptual semantic representations (see Table 1), while hierarchical patterns concentrate on the extraction of taxonomical links, following the work of [Hearst, 92], but based on the dependency formalism. Conceptual patterns are organized into a hierarchy and the extraction process tries to extract the most specific patterns first before going up in the hierarchy. For instance, the pattern "nsubj-dobj-xcomp" which is deeper in the hierarchy than the patterns "nsubj-dobj" and "nsubj-xcomp" should be first tested for instantiation in the current sentence. If a pattern is instantiated, then all its parents in the hierarchy are disregarded.

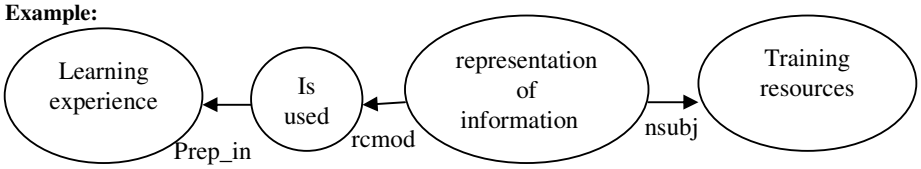
Table 1. An excerpt of the patterns grammar

Patterns

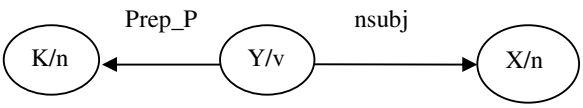


Transformations:
Join Z and P and creates a new link Z _P
Create predicate Z_P (X, K)

Example:

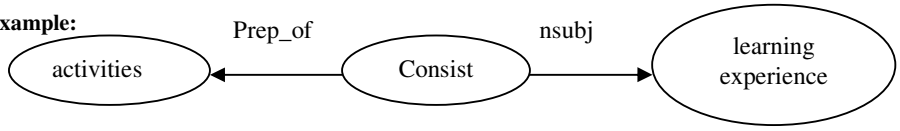


Semantic Analysis:
Is_used_in (training resource, learning experience)

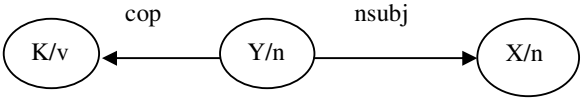


Transformations:
Join Y and P and creates a new link Y _P
Create predicate Y_P (X, K)

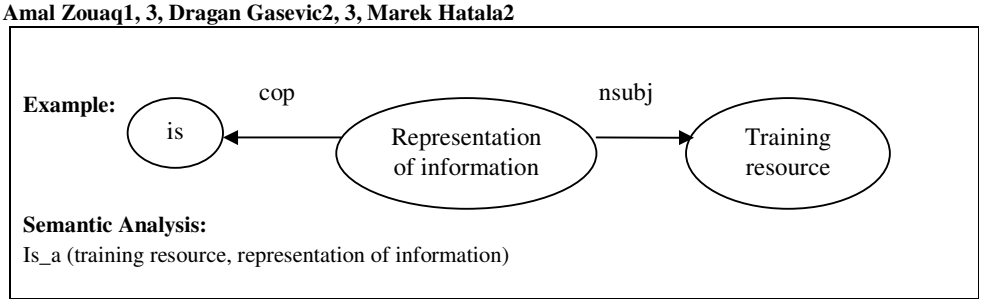
Example:



Semantic Analysis:
Consist_of(learning experience, activities)



Transformations:
Create predicate K (X, Y)



Patterns rely on transformations based mainly on the four generic operations:

- **Node fusion**: results into the aggregation of some nodes to form compound domain terms or relations. For instance, in the previous example, the dependency *nn(Specification-15, IMS-12)* would lead to the creation of one node *IMS Specification*;
- **Link/node fusion**: results into the creation of links labels from the fusion of some nodes and links. For instance, in the sentence learning experience consist of activities presented in Table 1, there is a link (*prep-of*) and a node (*consist*) fusion operation that creates the relation label "*consist_of*";
- **Link creation**: results into the creation of new links with specific labels (such as attribute or is-a links) mapped to some syntactic categories (such as prepositions, appositives, etc.) For example, in the sentence *the user's ID…,* there is a data type attribute in the form of a possessive attribute "*poss_attr*" that is created between the term *user* and the term *ID*.
- **Link copy**: in case of conjunctions, this operation is used to distribute the input/output links of the main conjunction term to the following ones, resulting into a distributive interpretation of the conjunction. In our example sentence, all the dependency relations linked to the word "SCORM-1" are duplicated and also linked to the word "specification-6" which is linked to "SCORM-1" in the original representation through a conjunction *and*.

Based on these patterns, OntoCmaps offers two kinds of tools: 1) a *document semantic parser* which highlights the identified patterns and the obtained representations sentence by sentence; and 2) a *corpus semantic parser* which performs the semantic analysis on the whole corpus, highlights the identified patterns and the obtained representations document by document (Figure 1) and performs the integration of the extracted representations. In figure 1, we can see the corpus on the left hand-side, the selected document in the center, and the list of extracted patterns, sentence by sentence in the list below the document.

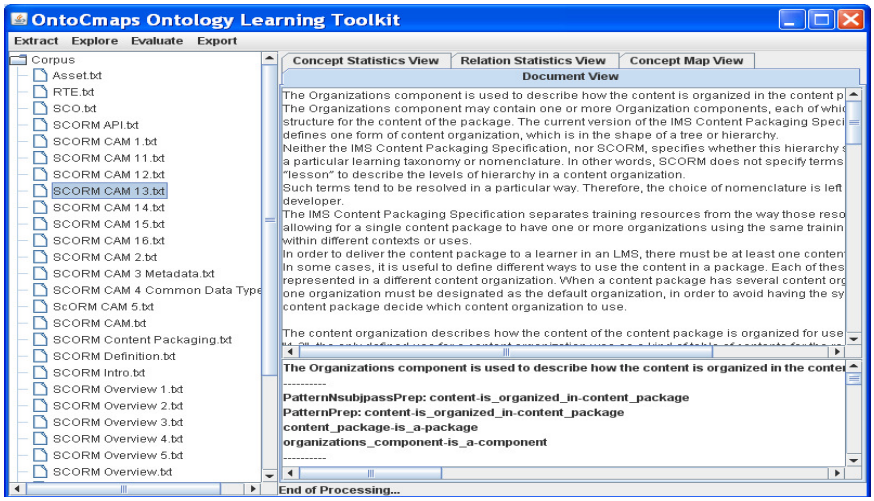


Figure 1. The OntoCmaps Tool

In this **integration phase**, concept maps are then created by performing basic term disambiguation using mainly: i) stemming such as mapping *assets* and *asset* to the same root *asset* and aggregating them in the same

concept map, and mapping the relations "launch", "can launch", "launching", and "launches" to the same relation); or; ii) basic synonym detection where synonyms resulting from abbreviation relations, such as SCORM runtime environment and SCORM RTE, are considered as the same concept and their relations belong to the same concept map. These two steps eventually result into concept maps around domain terms. There is also a co-reference resolution that is performed during the extraction phase in the execution of the rules associated with the patterns. For example, in the sentence: *assets are representations that do not call the SCORM API*, the co-reference resolution creates a relation between the term "*assets*" and the term "*SCORM API*" while the grammatical representation links the term "*representations*" to the term "*SCORM API*". The integration phase thus benefits transparently from this co-reference resolution. Of course, dealing with a homogenous domain corpus may greatly reduce the difficulty of this aggregation. In this work, we assume that the corpus is about the same domain.

The number of obtained concept maps after the integration may differ depending on the corpus and the detected patterns. If the corpus is well-chosen, then it is likely that OntoCmaps will produce one big connected graph with possibly few disconnected components. In fact, the well-chosen source corpus is an essential step in ontology learning [Brewster, 2008]. The corpus should encompass a majority of ontological statements (e.g., definitions or examples of concepts) and very few factual statements (e.g., named entities). These types of texts come mainly from academic textbooks or encyclopedias that are meant to be used by students for *learning a domain*. Similarly, an ontology learning system should also rely on these types of resources to learn the initial domain ontology, as we have previously advocated and empirically validated in [Zouaq & Nkambou, 2009]. Examples of appropriate sources would be a specific domain course or Wikipedia pages on some domain. However, their exact content should be manually checked to ascertain the existence of **ontological statements**, which is not the case for all Wikipedia entries). The Web can then be considered as a valuable source for extending the ontology.

4 Filtering in OntoCmaps

The third and last phase for learning the domain ontology is the **filtering phase**, which aims at ranking the obtained concept maps (domain terms, taxonomical links, and conceptual links). The proposed architecture (Fig. 2) is independent of a particular tool or framework. In fact, here we present our results as the filtering step of our tool OntoCmaps, but the process can be applied with any set of concept maps from a given domain. In this generic perspective, the input to the ranking step (shaded in Figure 2) is composed of domain specific concept maps that can originate from two sources: (1) emerging from domain corpora (texts) using a tool such as OntoCmaps, or (2) existing domain concepts maps from repositories. The output is a list of important concepts and relationships. Figure 2 shows the proposed generic pipeline as well as the tools used to support the discussed process (given on the right hand side of the figure).

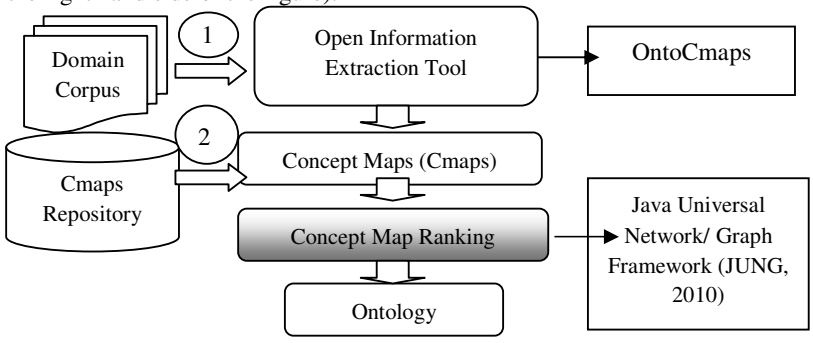


Figure 2. Conceptual Architecture

The necessity of filtering might be better explained through an example. In the previous example, we had the following extracted semantic relations:

1. Is_a(SCORM, application profiles of the IMS Content Packaging Specification)
2. Is_a(IMS SS Specification, application profiles of the IMS Content Packaging Specification)

3. Is_a(couple of restrictions, couple)
4. Is_a(profiles of the IMS Content Packaging Specification, profile)

We can notice that relations 1, 2 and 4 may be considered as "important", but that relation 3 may not be of interest to the domain because it is too general, and it should be filtered out. Other relations or terms may be wrong due to improper syntactic analysis or semantic analysis and should also be detected and assigned a very low rank.

Now the question is what kind of filtering measures are the best suited to perform this ranking.

4.1 Hypotheses

The approach presented in this paper is an experimental study that relies on a set of hypotheses to rank terms and relationships based on the structure of concept maps. As previously shown, these maps are constructed from terms and relationships between terms that are discovered in domain corpora or that are built manually (Figure 2). To apply graph-based metrics, terms of concept maps are considered graph nodes, and relationships are edges. Our assumption is that graphs or concept maps should exhibit some structural characteristics that may reflect the importance of nodes and relationships to the domain.

We propose a set of hypotheses that draw their roots mainly from the notion of centrality [Borgatti & Everett, 2006], which is essential in the analysis of networks. The hypotheses are based on metrics often used in social network analysis, namely the Degree centrality, the Betweenness centrality and the Eigen-vector centrality (PageRank, Hits). Thus, our hypotheses state that:

H1. The importance of a term may be a function of the number of relations that start from and end at the term. This can be measured using the Degree of a node, which is computed based on the number of edges that are incident to that node.

H2. The importance of a term may be a function of its centrality to the graph. This can be measured using Betweenness centrality. The betweenness centrality of a node A can be computed by the ratio of shortest paths through the node A connecting all pairs of other nodes in the graph.

H3. The importance of a term may be a function of the number of important terms that point to it. This can be measured using the Page Rank of a node, which is based on the number of links that point to the node, while taking into account the importance of the source nodes of these links. This is also related to the authorities detected through the Hits algorithm [Kleinberg, 99].

H4. The importance of a relationship may be a function of its source and destination concepts. Here, the important relationships are those which occur between two important concepts.

H5. The importance of a relationship may be a function of its centrality to the graph. Betweenness centrality can also be used to measure the centrality of a given edge.

All these hypotheses were tested within the OntoCmaps tool. It must be noted that the graph on which the metrics are applied is built using some pre-filtering mechanisms. For instance, OntoCmaps neglects extractions that contain stop words, tests if the arguments of a relation are nominal, and keeps only certain types of relations (verbal, equivalent, hierarchical and instance) in the graph for the filtering step.

4.2 Filtering Important Terms

A number of metrics from graph theory and from information retrieval were used to filter important terms and thus promote them as potential concepts in the domain ontology.

4.2.1 Metrics from Graph Theory

All the metrics from graph theory were computed using the JUNG framework [JUNG, 2010], which is a software library that enables the manipulation, analysis, filtering and visualization of data as graphs. In particular, JUNG offers a number of ranking algorithms that measure the influence, authority or centrality of a given node or relation in a graph. Since our goal was to measure the importance of terms, we chose four ranking algorithms namely:

- Betweenness centrality, which assign each node a value that is derived from the number of shortest paths that pass through it;
- The well-known PageRank algorithm [Brin & Page, 98];
- The Hits algorithm which ranks nodes according to the importance of hubs and authorities [Kleinber, 99];
- The Degree centrality which identifies the number of edges from and to a given node.

The four metrics: Degree, Betweenness, PageRank and Hits were computed as follows:

Given a graph G= (V, E) where the set of nodes (V) is represented by terms and the set of edges (E) is represented by labeled relationships between terms, and given a particular term *t*:

Degree (t) = the number of edges incident to t
Betweenness (t) = the ratio of shortest paths between any two terms that contain t
PageRank (t) = the fraction of time spent visiting t [JUNG, 2010]
Hits(t) = the authority score of t, which is related to the score of hubs that point to it

JUNG [JUNG, 2010] defines the PageRank of a node as the fraction of time spent at that node relative to all other nodes. In order to compute this fraction, the graph is transformed into a first-order Markov chain where the transition probability of going from node *u* to node *v* is calculated using the following formula [JUNG, 2010]:

$$PageRank(u)=(1-alpha)*[1/outdegree(u)] + alpha*(1/|V|)$$

where *alpha* is a parameter typically set between 0.1 and 0.2 and |V| is the number of vertices in the graph. Once the Markov chain is created, the stationary probability of being at each node (state) is computed using an iterative method.

All the metrics are normalized to be in the range [0-1] by dividing each value by the maximum value in the graph.

4.2.2 Standard Information Retrieval Metrics

In addition to the metrics from graph theory, we computed TF-IDF and term frequency (TF), two well-known metrics in information retrieval, as well as a random metric, which picks up important terms and relationships randomly. The TF-IDF metric is computed as follows:

TF_{ij} = *the number of occurrences of the term i/sum of the number of occurrences of all terms in document j*
$IDF_i = log$ (*|D|/number of documents containing the term i)*
Where |D| is the number of documents in the corpus
$TF\text{-}IDF_{ij} = TF_{ij} * idf_i$

The term frequency metric is computed by calculating the frequency of each domain term in the whole corpus and normalizing it using the maximum frequency. In OntoCmaps, the metrics TF-IDF and TF are calculated on the whole corpus, which includes compound domain terms, i.e. terms that emerge from the semantic analysis of texts through aggregation operations and filtered from stop words and certain parts-of-speech. For example, verbs cannot be considered as domain terms. This pre-filtering already enhances the precision of both metrics.

4.2.3 Voting Schemes

Using the graph theory metrics defined in Section 4.2.1, we defined a number of voting schemes with the aim of improving the precision of filtering as follows.

The **Intersection voting scheme**: in this scheme, a candidate term is considered as an important term if it is a candidate term for all four metrics (Degree, Betweenness, Hits and Page Rank). The voting scheme creates a set of terms TVoted where**:**

$$TVoted = TDegree \cap TBetweenness \cap TPageRank \cap THits$$

The **Majority voting scheme** recognizes a term as an important one if it is promoted by at least three metrics out of four, that is, it is a relaxed voting scheme compared to *TVoted*.

The **Weighted voting scheme** assigns various weights to the four metrics and computes various ranked lists of terms based on the values returned by each individual metric and its weight in the voting scheme. The general formula for the voting scheme is:

$$w1*TBetweenness + w2*TPageRank + w3*TDegree+w4*THits$$
$$Where\ w1+w2+w3+w4=1$$

An example of a Weighted voting scheme would be: *0.3*TBetweenness + 0.2*TPageRank + 0.4*TDegree + 0.1*THits*.

For all the three voting schemes, each important term is assigned a weight based on the combination of weights (addition and normalization) computed by each individual metric. For example, given the weights *w1*, *w2, w3* and *w4* assigned by each metric, the Majority voting scheme would assign a weight of (*w1+w2+w3+w4/4*).

Additionally, once important terms are computed using the voting scheme, the list of these terms is enriched by the following components: if an important term is part of a taxonomical link (as a child or as a parent) extracted using OntoCmaps, then its ancestor or descendant is added as an important term even if it was not selected by the voting scheme. We apply this rule to increase the number of important terms involved in taxonomical relationships. In fact, these relationships are very important for building a conceptual model and reasoning. This rule has also an impact on the rating of relationships by allowing, for instance, the selection of taxonomical links between important terms. (See the first measure for rating relationship importance below).

4.3 Filtering Important Relationships

Similar to the term weighting schemes, we also established a number of measures to rate the importance of relationships:

1 The first measure consists of selecting all the relationships that occur between important terms (determined through the three voting schemes) as important relationships. This constitutes our voting schemes for relationships. For example, an Intersection voting scheme for relationships will select all the relationships between the concepts identified by the Intersection voting scheme defined in section 4.2.3;

2 The second measure ranks relationships based on Betweenness centrality, where candidate relationships are chosen if their centrality is greater or equal than a threshold;

3 The third measure is based on assigning frequencies of co-occurrence weights based on the Dice coefficient (below), a standard measure for semantic relatedness;

4 The last measure is the Point-wise Mutual Information, which is computed based on the Google search engine.

The first two measures are based on the graph structural characteristics and can be considered as evidences based on the corpus from which the graphs emerge. The third measure is also based on the corpus, but it ranks relationships using the **Frequency of co-occurrence** of the nodes involved in the relationships. In this case, the importance of a relation *r* between S (Source) and D (Destination) is calculated by using the following formula:

$$Dice(S, D) = 2*F(S, D) / F(S) + F(D)$$

where:

- F(S, D) is the number of co-occurrences of S and D in a given context (here, an extracted relation).
- F(S) is the frequency of occurrence of S in the corpus.
- F(D) is the frequency of occurrence of D in the corpus.

Again, the selected relationships are those whose Dice coefficient is greater than or equal to the chosen threshold.

Finally, the last measure Point-wise Mutual Information (PMI) is based on the Web (Google) as a source of evidence. This metric is normally used to measure the semantic relatedness between two terms and it is used in OntoCmaps for computing the weight or probability of each extracted relationship. In our experiments, we relied on the Measure of Semantic Relatedness Server [Veksler et al., 2007] to calculate the PMI using the PMI-G met-

ric (PMI based on Google).

4.4 Thresholds

Determining a threshold is an important step that is generally based on experimentations. We decided to test various thresholds above which the ranked lists' elements would be considered as important. These thresholds are applied on the Betweenness, PageRank, Degree, Hits, TF and TF-IDF metrics for concepts and on the Betweenness, Frequency of co-occurrence and PMI-G metrics for relationships.

- The **Mean Threshold**: A term must have a measure (i.e., Degree, PageRank, Betweenness, Hits, TF-IDF and TF) greater than or equal to the mean value of the current metric to be considered as a candidate term. That is, for the six metrics, we create the following sets of terms TDegree, TBetweenness, TPageRank, THits, TTF-IDF and TTF where each of the sets contains terms that pass the threshold based on the mean values. The same threshold is also applied on Betweenness relationships, Frequency of co-occurrence and PMI-G metrics. The idea behind the mean value is to retain only the nodes that are already quite important instead of experimentally defining thresholds that may change from one corpus to another. Considering the mean value as a threshold assumes that less than a half of the extracted terms are important, which might be too restrictive. However, using this mean threshold is relevant in this paper given that our goal is to identify the most precise metrics.
- The **Percentage Threshold:** A given percentage of the ranked concepts and relationships lists is extracted and compared against a gold standard. Here we experimented with four percentages: 100%, 70%, 50%, and 30%.
- The **First Ranks Threshold**: A given number of elements is selected in the ranked concepts and relationships lists and compared against the gold standard.

5 Evaluation

The evaluation of our approach can be divided into two main experiments: the evaluation of the graph-based metrics against standard metrics and random baselines, and the evaluation of the ontology learning as a whole, with a comparison with the output of Text2Onto [Cimiano & Volker, 2005]. In both cases, this necessitates a gold standard. Text2Onto was chosen as the main state-of-the-art and freely available ontology learning tool. Other alternatives would have been: i) Text-To-Onto [Maedche &Volz, 2001], which is the ancestor of Text2Onto and which is the reason why we preferred the use of the more recent version Text2Onto; and, ii) OntoGen [Fortuna et al., 2004], but whose concept definition (classes of similar items) would have made the comparison of both outputs much more difficult for the domain expert and would have necessitated an adaptation of both outputs to one gold standard, thus leading to further complications. Additionally, previous experiments in our research groups showed that OntoGen generated ontologies were perceived by users as worse than those generated by Text2Onto [Hatala et al., 2009].

5.1 Description of the Experiment

5.1.1 Gold Standard Creation

Due to the well-known problem of evaluating the ontology learning task and lack of evaluation corpora, we decided to build our own corpus and gold standard[1]. We used a corpus of 30 000 words on the SCORM standard which was extracted from the SCORM manuals [SCORM, 2010] and which was previously used in another project [Zouaq & Nkambou, 2009]. The gold standard ontology was extracted in a two phase process designed to give equal chances to two tools: OntoCmaps and Text2Onto [Cimiano & Volker, 2005]. In the first phase, we

[1] Available at http://azouaq.athabascau.ca/corpus/SCORM/Scorm.zip

ran both the OntoCmaps and Text2Onto tools on the corpus and generated a domain ontology from either tool. The objective was to create a single ontology from both outputs. We generated the OntoCmaps initial domain ontology without any filtering, that is, all the extracted terms and relations were exported without any filtering and ranking. In Text2Onto, we had to choose a number of algorithms to extract the domain ontology (see Figure 3) with an average combiner when different algorithms were used to extract the same ontology layer. For instance, we used the average combiner for extracting concepts based on TF-IDF, entropy, Relative Term Frequency (RTF) and Example-based extractions. To keep the comparison as fair as possible, we used all the available algorithms for each ontology layer (with an average combiner) except the algorithms that rely on external resources (such as WordNet).

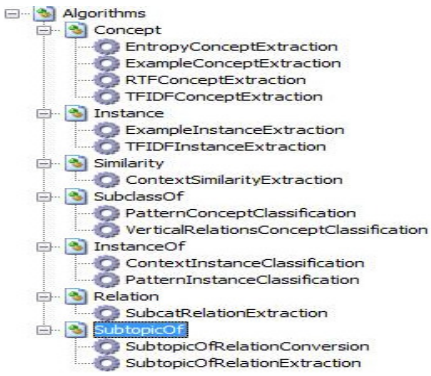


Figure 3. The Text2Onto algorithms

We then automatically merged both ontologies into a single one, by automatically performing some basic alignments (e.g., the same term or relation extracted by Text2Onto and OntoCmaps was exported only once and items extracted by one of the tools only were added to the ontology with no further checking) and we ended up with a single large ontology. In the second phase, this ontology was presented to a domain expert. Not only was the expert familiar with the domain, but also with ontologies and with both tools (OntoCmaps and Text2Onto). In order to obtain the final gold standard, the expert had the task to revise the merged ontology, namely by deleting erroneous concepts and hierarchical and conceptual relationships, and by adding hierarchical relationships, which should have been normally generated by the tools. The expert did not have to add any conceptual relationship not already included in the ontology.

The statistics for the generated ontologies are shown in Table 2. The very low number of relationships extracted by Text2Onto by comparison with OntoCmaps can be noted and shows how little semantics can be extracted using only shallow NLP and statistical methods.

Table 2. Number of extracted terms and relationships

Number of	**Text2Onto**	**OntoCmaps**	**Merged Ontology**
Primitive classes	1457	2393	2594
Defined classes	0	64	64
Conceptual relationships	85	1707	1731
Hierarchical relationships	432	1610	1610

Table 3 shows the Gold standard ontology statistics after the expert evaluation and cleanup of the merged ontology.

Table 3. The gold standard statistics

Number of	**Gold Standard**
Primitive classes	1384
Defined classes	64

Conceptual relationships	728
Hierarchical relationships	1345

Next, the ontology was exported as an Excel spreadsheet containing gold standard concepts and relationships (defined classes were ignored).

5.1.2 Gold Standard Evaluation

The gold standard was created by only one domain expert due to

- The difficulty of finding domain experts ready to model the information in domain corpora (which implies reading them) in the form of a domain ontology;
- The associated time and financial cost of such a process.

In fact, the ontology learning community suffers from the lack of gold standards (domain ontologies) and their corresponding corpora. This claim is best validated by examining the significant literature in the area, where one can hardly find two different studies that used the same evaluation materials. There are a number of domain ontologies that exist, but they are generally not linked to a precise document collection and they result from the collective effort of domain experts based on their experience of the domain and their background knowledge. However, it is not fair to compare an automatically learned ontology based on a specific corpus with such expert ontology as the learned ontology will depend heavily on the corpus from which it is extracted.

To counterbalance the bias that may be introduced by relying on a unique domain expert, in our study, we performed user tests to evaluate the correctness of the previously produced gold standard (c.f. Sect. 5.1.1). We randomly extracted concepts and their corresponding conceptual and taxonomical relationships from the gold standard and exported them in Excel worksheets. The worksheets were then sent together with the domain corpus and the obtained gold standard ontology to 11 users from Athabasca University, Simon Fraser University, the University of Belgrade, and the University of Lugano. The users were university professors (3), postdoctoral researchers (2), and PhD (5) and master's (1) students. The users were instructed to evaluate their ontology subset by reading the domain corpus and/or having a look to the global ontology. Each user had a *distinct set of items (no duplicated items)* composed of 20 concepts and all their conceptual and taxonomical relationships.

Table 4 shows the number of concepts, taxonomical relationships and conceptual relationships that were randomly extracted and evaluated by our users. As the table indicates, almost 29% of the entire gold standard was evaluated by users. This size of the sample and the fact that the sample evaluated by the user was selected randomly can provide us with solid evidence that the results of the user evaluation of the sample can be generalized to the entire gold standard.

Table 4. The number and percentage of randomly extracted items for the user evaluation of the gold standard

Type of items in the gold standard	Total number of the extracted items from the gold standard	Percentage of the extracted items from the gold standard (%)
Concepts	209	15.10
Conceptual relationships	182	25.00
Hierarchical relationships	617	45.87
Overall (all three item types) average percentage		28.66

Each user was asked to evaluate the correctness of his subset of items through the following coding scheme:

- **Important**: The users were asked to choose this option if they considered that the item should be present in an ontology about the SCORM standard;
- **Understandable**: The users were asked to choose this option if they understood why this item could be chosen by an automatic extraction system, but they would not necessarily have included it while designing an ontology about SCORM manually;
- **Invalid**: The users were asked to choose this option if they considered that the item should not be a part of the SCORM ontology or they considered it as erroneous.

Table 5 shows the results of the users' evaluation in detail and also summarizes the data under the categories

"Accepted" and "Rejected" where *accepted* is constituted by the percentage of important and understandable items and *rejected* is constituted by invalid items.

Table 5. Percentage of each coding scheme category for concepts, conceptual relationships and taxonomical relationships

	Accepted			**Rejected**
	Important (1)	**Understandable (2)**	**(1) + (2)**	**Invalid**
Concepts	58.85	33.49	92.35	7.65
Conceptual relationships	38.46	41.76	80.22	19.78
Taxonomical relationships	46.19	42.79	88.98	11.02

We can notice that more than 90% of the concepts and more than 80% of the relationships were accepted by the participants of our evaluation. Based on a random selection of items, this user-based evaluation confirms that the obtained gold standard could be considered *valid* (although *not perfect*) for evaluating the effectiveness of an ontology learning system. Although there is no general consensus about the interpretation of the level of inter-rater agreement in computational linguistics [Artstein & Poesio, 2008], our results can be interpreted almost prefect according to some inter-rater agreement approaches [Krippendorff, 2004].

We also analyzed manually the answers of the users and we noticed some problems without which the percentage of accepted items would have been even higher. In fact, the items included in the ontology are stemmed, which means that only the root of a word constitutes the class name. For example, "packag" is a class of our ontology. However, there are a number of labels that are associated with each root such as the labels "packaging", "package," and "packages" which are associated to the concept "packag". For the sake of the experiment, we wanted to avoid presenting the class names (roots) which could confuse users, which were not used to work with and understand stems. Thus, the worksheet generator only extracted the first class label of each class, in our example "packaging". Unfortunately, this resulted into another misunderstanding since a number of users declared as invalid some concepts and relationships whose labels were not conjugated or grammatically correct. For example, the taxonomical relation: "***Shareable content object reference model-is a-specific***" was declared invalid by one of the users while the concept "specific" designates here the term "specification", which is clearly a valid and important relationship for the SCORM domain. This is an issue that has to be solved in future evaluations.

5.1.3 Measures Used in and Objectives of the Evaluations

Once the gold standard was validated by the users, the various metrics were evaluated against the gold standard using three well-known measures of information retrieval:

Precision = items the metric identified correctly / total number of items generated by the metric
Recall = items the metric identified correctly / total number of relevant items (which the metric should have identified)

$$F\text{-}Measure = 2 * ((precision * recall) / (precision + recall))$$

The objectives of the experiment were:
- To assess how well a given metric performs under various threshold;
- To identify the best metrics according to a given measure (precision, recall and F-measure) for concept and relationship ranking;
- To compare metrics from graph theory with two baselines: a naïve baseline and traditional text-based metrics baseline., i.e. TF-IDF and TF for terms and Frequency of co-occurrence and PMI-G for relationships;
- To compare the results of OntoCmaps with those of Text2Onto.

We then ran again the OntoCmaps tool on the corpus and we exported the system recommended concepts and relationships (hierarchical and conceptual) for each metric into an Excel spreadsheet. We repeated this operation for each threshold: the mean threshold, the percentage threshold (100%, 70%, 50%, and 30%) and the first ranks thresholds (100, 200 and 400, i.e., first 100^{th}, 200^{th}, and 400^{th} ranked items).

The results are presented below for each threshold with a particular emphasis on the mean threshold, which does not depend on experimental tests and can be applied regardless of the corpus size and the number of extracted terms and relations.

5.2 Mean Threshold Results

We obtained a ranking of terms based on the metrics: Hits, Degree, PageRank, Betweenness, Intersection Voting Scheme, Majority voting scheme, Weighted voting scheme and the three baselines Random Concepts, TF and TF-IDF. Note that many Weighted voting schemes were tested, but we present here only the best Weighted voting scheme statistics and results. Here by best voting schemes, we mean voting schemes that obtained the best precision and/or F-measure. Tables 7, 8 and 9 show the number of terms that were selected by each metric and by the random baseline as important terms (concepts).

Table 7. Number of identified concepts according to all the graph-based metrics

Hits	Degree	Page Rank	Betweenness
349	525	475	293

Table 8. Number of identified concepts according to all the baselines

TF	TF-IDF	Random baseline
382	537	712

Table 9. Number of identified concepts according to all the voting scheme metrics

Intersection VS	Majority VS	Weighted VS (Bet 0.0; Prank 0.0; Degree 0.8; Hits 0.2)
692	1002	821

Regarding relationships, four metrics were tested as previously indicated: a measure that selects a relationship if it occurs between two important terms identified by a voting scheme, Betweenness centrality, and the three baselines Frequency of co-occurrence, PMI-G and random relationships. Tables 10 and 11 display the number of extracted relationships for each metric. The first two metrics are based on graph theory whereas the last two metrics (Frequency of co-occurrence and PMI-G) are standard metrics commonly used to rate relationships in NLP tasks.

Table 10. Number of identified relationships according to the graph-based metric and baselines metrics

Betweenness	Frequency of co-occurrence	PMI-G	Random baseline
449	454	905	563

Table 11. Number of identified relationships according to the voting scheme metrics

Intersection VS	Majority VS	Weighted VS (Bet 0.0; Prank 0.0; Degree 0.8; Hits 0.2)
452	745	547

5.2.1 Concepts

The following table shows the precision, recall and F-measure obtained for each metric. We can notice that the baselines (random and text-based) are all outperformed by the graph-based metrics. Surprisingly, the TF metric performs better than TF-IDF. This might be due to the fact that these two metrics are applied on the pre-filtered domain terms and not on the whole corpus as this is usually done.

Table 12. Metrics results for concept identification

		Precision	Recall	F-measure
Graph-based metrics	Betweenness concept	74.40	14.70	24.56
	Hits Concept	73.06	17.20	27.85
	PageRank concept	70.10	22.46	34.03

	Degree concept	71.23	25.23	37.26
Baseline metrics	TF-IDF concept	44.50	16.12	23.67
	TF concept	60.20	15.51	24.67
	Random concept	53.08	25.50	34.45
Voting schemes	Intersection concept	81.79	38.19	52.06
	Majority concept	78.64	53.17	**63.44**
	WVS 0.0;0.0;0.8;0.2	**82.82**	45.88	59.05

We can also note that all the voting scheme metrics outperformed the base (i.e., graph-based) metrics in terms of precision, recall and F-measure. The Majority voting scheme has the best F-measure among the voting schemes. Finally, the somewhat good performance of the random baseline in terms of F-measure might also be due to the random choice of concepts among pre-filtered domain terms. However, its precision is very low by comparison with graph-based metrics.

5.2.2 Conceptual Relationships

Regarding conceptual relationships, the text-based metrics (Frequency of co-occurrence and PMI-G) and the Betweenness metric are all outperformed by our voting schemes (Hypothesis H4), that is, by the metrics that identified important relationships as those occurring between concepts selected by a given voting scheme. Especially, we can notice that Intersection conceptual relationships obtain the highest precision, but that the Majority voting scheme obtains the highest recall and F-measure.

Table 13. Metrics results for conceptual relationships

		Precision	**Recall**	**F-measure**
Graph-based metric	Betweenness Conceptual Relation	48.32	30.13	37.12
Baseline metrics	PMI-G Conceptual Relation	41.54	52.22	46.27
	Co-occurrence Conceptual Relation	45.81	28.88	35.43
	Random Conceptual Relation	39.78	31.11	34.91
Voting schemes	Intersection Conceptual Relation	**60.84**	38.19	46.92
	Majority Conceptual Relation	50.73	52.50	**51.60**
	WVS Conceptual Relation 0.0;0.2;0.0;0.8	59.23	45.00	51.14

5.2.3 Hierarchical Relationships

As far as taxonomical links are concerned, we can notice that random hierarchical links already obtain a quite good precision. The Majority voting scheme obtains the best results in terms of F-measure while a Weighted voting scheme slightly outperforms the Intersection voting scheme in terms of precision. We can also note that the recall is much better in the voting schemes due to our rule that adds hierarchical relationships' concepts if one of them is selected by a voting scheme.

Table 14. Metrics results for hierarchical relationships

		Precision	**Recall**	**F-measure**
Graph-based metric	Betweenness Hierarchical Relations	64.51	1.48	2.90
Baseline metrics	PMI-G Hierarchical Relations	66.99	5.12	9.52
	Co-occurrence Hierarchical Relations	68.00	1.26	2.47
	Random Hierarchical Relations	65.78	3.71	7.03
Voting	Intersection Hierarchical Relations	79.84	37.66	51.18

schemes	Majority Hierarchical Relations	77.64	49.55	**60.49**
	WVS Hierarchical Relations 0.1;0.8;0.1;0.0	**81.04**	45.09	57.94

To increase the recall in all metrics, we added all the hierarchical relationships to all the metrics. This resulted in a precision/recall/F-measure of 66.49/75.33/70.63. This seems to indicate that filtering hierarchical relations might not be a promising option, since a very good proportion of these links is accurate. This fact has also been confirmed by the expert in charge of creating the gold standard.

In the following sections, we will focus on concepts and conceptual relations.

5.3 Percentage Threshold Results

In the percentage threshold, a given percentage (100%, 70%, 50% and 30%) of the ranked terms and relationships was selected for each metric and the results compared for each percentage. The 100% threshold means that no filtering is performed and that all the metrics accept all the extractions.

5.3.1 Concepts

Figures 4 and 5 show the results obtained for the 50% and 70% thresholds. As can be noticed in both figures, the Intersection and Majority voting schemes obtain the best results.

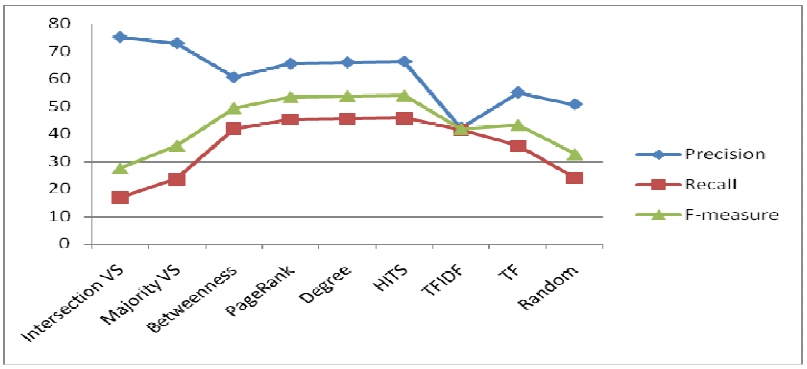


Figure 4. Metric Results with the 50% threshold

In the 70% threshold (Figure 5), we can note that the graph-based metrics follow a similar curve with almost the same results for precision, recall and F-measure. Again precision is the best when the Intersection and Majority voting schemes are used and the baselines (TF, TF-IDF, Random) are always outperformed by the graph-based metrics and by the voting schemes.

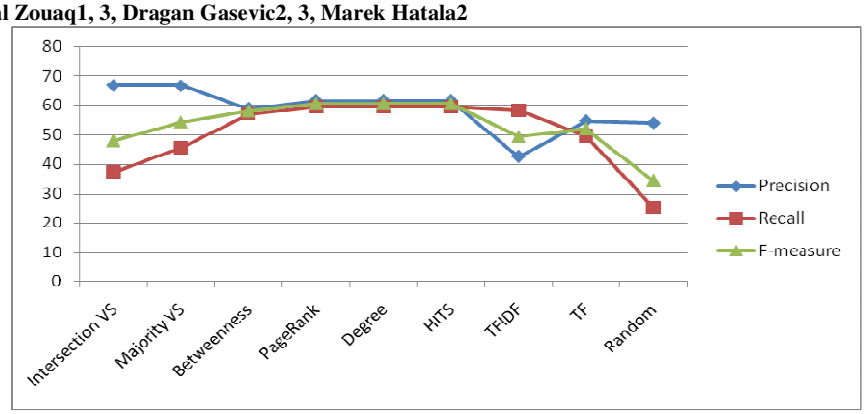


Figure 5. Metric Results with the 70% threshold

In the 100% threshold (no filtering), the voting scheme metrics obtain a precision/recall/F-measure of 58.12/88.59/70.19. We can thus notice that filtering contributes to a higher precision for voting schemes meaning that we were able to increase precision to as much as 66% in the 70% threshold and 75% in the 50% threshold. Finally the best results in terms of precision are obtained by the 30% threshold (>83%) as it is the most selective threshold.

5.3.2 Conceptual relationships

Figure 6 shows the results of the 50% threshold for conceptual relationships. Here, we can notice again that the Majority voting scheme performs better than all the other metrics in terms of recall and F-measure, followed by the Intersection voting scheme which performs the best in terms of precision. The same results are observed for the 70% threshold in terms of recall and f-measure. The best recall and F-measure are always obtained by the Majority voting scheme in the 50% and 70% thresholds.

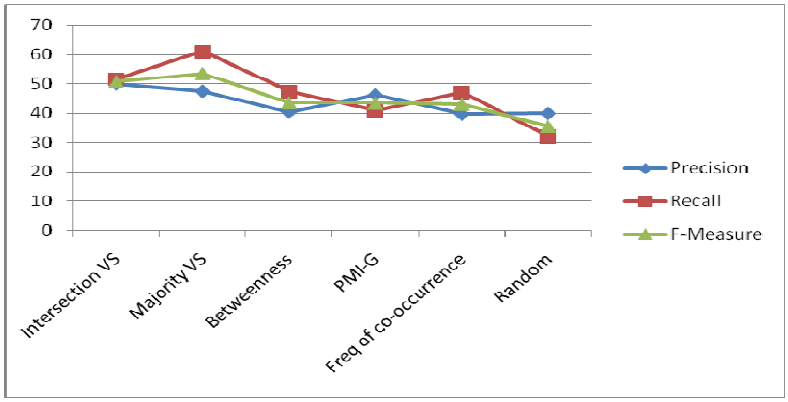


Figure 6. Metric Results for conceptual relationships with the 50% threshold

In the 100% threshold (no filtering), the voting scheme metrics obtained a precision/recall/F-measure of 39.87/86.94/54.67, meaning that we were able to increase precision to as much as 42% (70% threshold), 49% (50% threshold) and 62% (30% threshold).

5.4 First Ranks Threshold Results

Finally, the first ranks thresholds are calculated only in terms of precision. Over a given number of terms and relations, the experiment determines the number of items that are correctly identified by the metric. The results are shown in Table 15.

Table 15. First ranks threshold results

	Precision	First 100	First 200	First 400
Graph-based metrics	Betweenness concept	82.00	**80.00**	74.31
	PageRank concept	85.00	77.00	72.50
	Degree concept	**85.00**	**80.00**	73.25
	Hits Concept	79.00	75.00	72.83
Baseline metrics	TF-IDF concept	67.00	59.50	48.25
	TF concept	64.00	61.00	60.20
	Random concept	53.00	54.00	52.25
Voting schemes	Intersection concept	80.00	78.00	**80.50**
	Majority concept	84.00	79.00	75.50

Again, the graph-based metrics outperform the baselines in all the thresholds (100, 200, and 400). In the first 100 terms, the degree and PageRank metrics have the best precision. Regarding the degree, this is consistent with the fact that terms which have many relationships with others should be considered as important.

As we enlarge the set of items to be ranked (200 and 400), we can notice a drop in precision for all metrics (except the Intersection voting scheme, which improves its precision at the first 400 threshold), but also the growing importance of the Betweenness metric. In fact, as the set of considered concepts grows to 400, Betweenness has the best results among the graph-based metrics. Finally, the Intersection voting scheme obtains the best result in the biggest threshold (400), which is a logical consequence of noise introduction as the set of items grows.

As far as relationships are concerned, it is clear that the Intersection voting scheme is always the best (Figure 8), followed by the Majority voting scheme and the PMI-G metrics (up to the 200 relations). However, the PMI-G metric outperforms the Majority voting scheme at the 400 threshold.

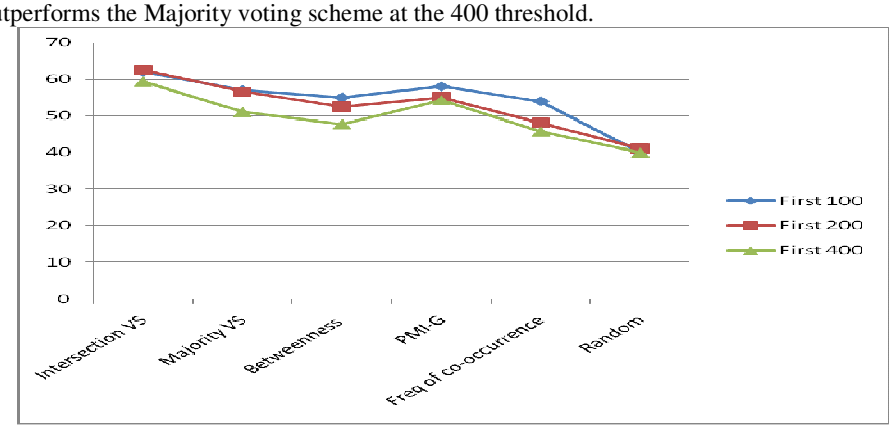


Figure 7. Metrics results for conceptual relationships with the first ranks threshold

5.5 Text2Onto Results

The final experiment was to the compare the results of the Text2Onto tool, which uses standard metrics for concepts and relationships ranking, with those presented above. In particular, Text2Onto implements a standard version of TF-IDF on the whole corpus and not on a pre-filtered version of the corpus as this is done in OntoCmaps. This enables us to compare also the "traditional" TF-IDF with our results.

The results of Text2Onto on the whole corpus are shown in Table 16 while the results by using the First ranks thresholds (100, 200 and 400) are presented in Table 17. We can conclude in both cases that they are far below those obtained by OntoCmaps in terms of concepts, hierarchical relations and conceptual relations.

Table 16. Text2Onto results on the whole corpus

	Precision	**Recall**
Concepts	31.71	25.16
Conceptual Relations	14.11	1.66
Hierarchical Relations	29.06	9.95

Table 4. Text2Onto results for the first rank thresholds

	100	**200**	**400**
Concepts	43.00	34.00	25.00
Conceptual Relations	14.28	14.28	14.28
Hierarchical Relations	30.00	27.50	28.50

5.6 Discussion

5.6.1 Ontology learning

At the global level of ontology learning, it is clear that our research yields promising results. In fact, by comparison with the results obtained by Text2Onto, OntoCmaps generates a richer ontology particularly in conceptual relationships. At all levels, the precision and F-measure obtained by Text2Onto are far below those obtained by OntoCmaps. We noticed that Text2Onto had difficulty in: i) building compound domain terms and relationships, while this is the strongest point of OntoCmaps, due to its deep semantic analysis, and ii) identifying many conceptual relationships, due to Text2Onto's shallow NLP techniques.

It must be noted that OntoCmaps only advocates the semi-automatic generation of a domain ontology, where the expert should play a central role to assess and validate the results. At the moment, OntoCmaps offers visualization capabilities at the corpus level to the domain expert, who is capable of visualizing a concept map related to a particular term and at the document level, where the extracted representations are highlighted sentence by sentence and shown in context. Another guidance of OntoCmaps is the weight that it assigns to all its elements using the filtering mechanism (see Section 5.6.2) and the presentation of ranked results to the user, thus helping him in deciding quickly of the most important elements. This functionality of the tool will be later evaluated through a usability study.

5.6.2 Filtering

One important objective of OntoCmaps is the ability to filter the extracted concepts and relationships by assigning probabilities to these items and filtering the noise that results from errors in the NLP process as well as from the unsupervised nature of the extraction.

Our results indicate that using metrics from graph theory may help identify important concepts (> 80% precision) and relationships (> 60% precision) in a more precise manner. By comparison with standard measures such as TF-IDF and TF for concepts, and Frequency of co-occurrences or PMI-G for relationships, graph-based metrics give better precision and overall F-measure results. Metrics from graph theory (**Hits (**as per **H3** from Section 4.2); **PageRank (H3)**; **Degree (H1)***;* **Betweenness (H2)**) are consistently giving better precision results than the random metric, or than standard measures such as TF-IDF or TF for concept identification. The degree metric (H1 hypothesis) is particularly promising for the most important terms that are linked to many others through conceptual and hierarchical relationships. The Degree metric's importance decreases when the set of considered items (terms and relationships) grows as shown in Table 15. This can be explained by the fact that there are also highly specific concepts that do not have a high degree but which might nevertheless be important. These concepts might then be captured by a voting scheme (Intersection and Majority).

Regarding relationships, our findings suggest that an **Intersection voting scheme** (*TVoted*) for concept identification is the most likely to give better precision results and that a **Majority voting scheme** is the most likely to give the best F-measure when compared to a Random baseline, Frequency of co-occurrences, Betweenness and PMI-G (i.e., H4 hypothesis in Sec. 4.2). However, the **PMI-G** semantic relatedness measure gives also an inter-

esting perspective with an F-measure slightly higher than the Intersection voting scheme one. The good performance of PMI-G, which is based on the Web and Google, probably is the fact that it takes into account the redundancy of information aspect [De boer et al., 2007], which has been discovered as very important for weighting information. It would be interesting to combine our voting scheme with the results of the PMI-G metric. However, PMI-G as it was used in this experiment only indicates the strength of the link between two terms, but does not indicate the correctness of the labeled relation between these two terms. One way to remedy this is to include the label in the calculation of the PMI-G between two terms. This will be tested in further experiments.

Although these results in terms of precision/recall are far from being perfect, they outperform the results of Text2Onto on the same corpus and they are still very reasonable especially if they are considered in a semi-automatic approach where the expert could adequately navigate the results helped with visualization capabilities. Generally, it is very difficult to obtain a high precision in this type of ontology learning task. For example, the experiment of [Brewster et al., 2009] on the animal behaviors domain obtained a precision of only 26% (with 73% recall).

In general, **Weighted voting schemes** are also promising, but they require many experimental tests which may hinder their potential benefits. Hierarchical relationships are generally accurate and do not require filtering. Moreover, erroneous hierarchical relationships might be detected easily and in a straightforward way by the expert.

These results confirm that using metrics from graph theory can be a promising way to extract important conceptual structures from free texts without recurring to an external knowledge resource.

5.6.3 Limitations

There are a number of limitations to our experiments. From an extraction perspective, as previously said, the whole NLP pipeline might propagate errors from one stage (POS tagging, dependency syntactic parsing, and semantic parsing) to the other. Evaluating the accuracy of each step might be an interesting avenue to explore especially by weighting the probability of the extraction patterns.

The accuracy of the whole NLP process might heavily depend on the corpus itself. In fact, as previously stated, the choice of an adequate corpus is a major step for the success of an ontology learning tool and this phase is particularly neglected in current state of the art. Due to the time required for building a gold standard (one month of full-time work was necessary in our case), the approach was tested on only one corpus. The effect of changing the domain might be interesting to test if the same metrics emerge as the best from one domain to the other. Our assumption and preliminary experiments is that domain change will not affect the results at the semantic analysis level since the approach is domain independent. However, the corpus should mainly be constituted of definitions and explanations about the domain able to be processed by our main patterns. Moreover, the accuracy of the dependency parses is a major issue for the success of the subsequent semantic analysis. Analyzing the performance of the Stanford parser on various domains might be very important to assess the overall performance of OntoCmaps.

Our patterns have been designed by taking into account the grammatical relationships used in the Stanford parser. Although other parsers should have similar relations, the exact terminology should be mapped to the Stanford one to be able to change the syntactic parser. One interesting avenue to explore would be to learn a syntax-semantic interface using this mapping in order to extract automatically the grammar for a new parser instead of building it manually.

Another limitation is that OntoCmaps is “only” able to extract information explicitly stated in the text. Very few relationships are inferred implicitly such as attributes coming from the preposition “of” or from possessives such as “the user’s name”. This is also a reason why the corpus selection should be considered as a major step of the ontology learning process and should mention explicitly as much information as possible.

From an evaluation perspective, it would also be interesting to assess the quality of the concept maps before filtering as well as the quality of the corpus from which they are extracted using formal metrics such as those used in information retrieval. Instead of choosing a particular threshold, we might decide to rank all extractions and let the expert decide the lowest ranks to be included. The mean used as a threshold for each metric might be too restrictive, which might also explain the very low obtained recalls with this threshold.

Finally, we are currently planning experiments that involve more than one domain expert in the development of a larger gold standard, and a detailed analysis of the results. We believe that our results are interesting and of-

fer a novel research result especially if we compare the performance of the proposed metrics with the performance of standard measures on the same corpus (TF-IDF, PMI-G, Frequency of co-occurrence) and with the performance of a random baseline.

6 Related Work

6.1 Ontology Learning

Ontology learning pursues the goal of creating formal models for the Semantic Web. We showed in this paper that this process can be decomposed into three phases: knowledge extraction, filtering and integration, and that the majority of the approaches rely mainly on shallow NLP, while deeper analysis methods might be needed, especially for conceptual relation learning and axiom extraction. In fact, the majority of the ontology learning approaches produce shallow ontologies mainly constituted of domain terms and hierarchical relations [Volker et al., 2008]. Approaches such as Text-to-Onto [Maedche &Volz, 2001], OntoGen [Fortuna et al., 2004], OntoLearn [Navigli & Velardi, 2004] and Text2Onto [Cimiano & Volker. 2005] all generate these types of ontologies. While these generated ontologies have been very useful for annotation purposes, and as a starting point for more formal structures, there are application domains that require more expressive ontologies [Volker et al., 2008] that enable reasoning. The work presented in this paper is one step in this direction especially with richer conceptual relationships between concepts and the ability to extract deeper semantic representations than what is proposed in the state of the art.

6.2 Open Information Extraction

Recent efforts in the knowledge extraction community has encouraged the development of open information extraction [Banko et al., 2007] [Etzioni et al., 2008]; that is, a text mining software tool that does not rely on predefined templates, but that is able to extract knowledge in an unsupervised way. However, open information extraction, in systems such as TextRunner [Banko et al., 2007] or Woe [Wu & Weld, 2010], are generally aimed at extracting factual knowledge rather than conceptual knowledge.

Other initiatives for building knowledge bases from texts exist, such as the "learning by reading" [Barker at al., 2007] challenge, NELL [Carlson et al., 2010] and Kleo [Kim & Porter, 2009], where a machine is supposed to grasp important knowledge from texts and develop a conceptual model. However, in many of these initiatives, the systems rely on a structured knowledge base to guide the extraction and populate the existing schema with instances. Moreover, this presupposes that the systems are guided in their extraction and know what they are supposed to extract at a fine-grained level, such as what is done in named entity recognition. To our knowledge, this paradigm of open information extraction has not been yet proposed in the ontology learning task.

6.3 Concepts and Relationships Learning from Texts

Traditionally, in statistical and machine learning approaches, finding concepts and relationships in texts has been performed by linking them to a semantic repository [Pantel & Pennacchiotti, 2008] and by estimating the probability of the relationships [Soderland & Mandhani, 2007] or concepts [Cimiano & Volker, 2005], [Maedche & Volz, 2001] based on standard measures from information retrieval, such as TF-IDF [Salton & Buckley, 1988] or C/NC value [Frantzi & Ananiadou, 1999]. Clustering methods [Pantel & Lin, 2002][Pantel & Pennacchiotti, 2008] [Soderland & Mandhani, 2007] have also been used to find some categories in the data that are then considered as concepts or patterns. There are also highly supervised machine learning approaches for learning concepts or specific relationships (e.g., synonyms, hyponyms, and causal), which lead to very accurate results, but which suffer from their dependence upon hand-labeled and domain dependent examples. Knowledge-based approaches often rely on WordNet to annotate the extracted data, hence linking it to a conceptual structure. For instance, Espresso [Pantel & Pennachiotti, 2008] uses a clustering technique based on the WordNet taxonomy to create reliable extraction patterns and assigns a WordNet sense to terms linked by relationships that hold. Kleo [Kim & Porter, 2009] also relies on an established knowledge base, the Component Library and WordNet, to as-

sign meaning to the extracted information.

One drawback of the aforementioned works is that they rely on a knowledge base or clustering examples. Due to the extensive effort required to build and maintain such knowledge structures, and due to the inadequacy of some of these structures (e.g., WordNet) to represent domain knowledge, it is our current position that semantics should emerge from texts without recurring to a predefined knowledge base as was proposed in this paper.

6.4 Metrics from Graph Theory

Metrics from graph theory have been widely used in social network analysis to discover for instance communities of users [Kleantheous & Dimitrova, 2008], to rank Actor-Concept-Instance networks [Mika, 2005], to provide visualization filters [Jia et al., 2008] or to rank already available ontologies [Patel et al., 2003] [Alani & Brewster, 2006] [Hoser at al., 2006] [Zouaq & Nkambou, 2009]. Other approaches have successfully used metrics from graph theory for word-sense disambiguation [Navigli & Lapata, 2010] or for topic identification [Coursey & Mihalcea, 2009] and they have been applied to ontologies for the purpose of analysis [Alani & Brewsterm 2006] [Hoser et al., 2006] *but not for filtering purposes*. Similarly, metrics of semantic relatedness have not been used, to the best of our knowledge, to rate relationships in ontology learning.

In general, probabilities and weights have been used to filter important terms in ontology learning approaches [Buitelaar & Cimiano, 2008], but they generally rely on measures such as TF-IDF or Frequency of co-occurrences rather than on the structural characteristics of graphs. To the best of our knowledge, there are very few, if any, attempts to exploit such graph-based metrics to filter the results of an information extraction system and to extract interesting concepts and relationships from graph structures. One proposal in this direction is our own work [Zouaq, 2008] [Zouaq & Nkambou, 2009], which identifies the out-degree of a term (the number of edges whose source is the term) as an indicator of the importance of the term. All the terms whose out-degree is greater than a given threshold are considered as concepts. However, there is no comparison, in that work, with other kinds of metrics such as Betweenness centrality, Degree or Page Rank to rate the effectiveness of each of these metrics for evaluating concept importance.

7 Conclusion

This paper presented an approach to learning a domain ontology in an open manner. The approach also addresses ranking and filtering relevant terms and relationships in concept maps using metrics from graph theory. The novelty of the approach is that 1) it uses deep semantic analysis methods and generates rich conceptual structures in the form of concept maps; 2) it relies on the inner structure of graphs to identify the important elements **without using any other knowledge source**. The other contribution is that we addressed the problem of filtering concepts and relationships with good precision and F-measure. Not only may our approach be beneficial for automatic extraction tools, but it may also for analysis of concept maps repositories, as well as for analysis of any graph-based representation of texts such as co-occurrence graphs. Our experiments showed that exploiting a voting scheme based on the metrics of PageRank, Hits, Betweenness and Degree provides a good concept identification precision. The other finding was that important relationships are better identified with the Intersection and Majority voting schemes. In general, the best metrics were always based on graph theory and centrality measures. The approach was assessed using a human evaluation and showed that a good overlap might be obtained between our system's results and the human results. We also compared our approach with Text2Onto and obtained better results.

As demonstrated in the paper, our approach might be interesting for identifying conceptual structures in general and for ontology learning and evolution in particular. In fact, one of the main difficulties for ontology acquisition from text is the correct extraction and identification of important concepts and relationships. The results presented in this paper suggest that using graph theory may be an interesting avenue to identify ontological classes and relationships (taxonomical links and properties with domain and range) with a higher degree of precision. This should help, for instance, users in building high quality ontologies using a semi-automatic process where an initial ontology design can be based on the investigated measures. To address the challenge, in future work, we plan to have user studies where such initial ontologies will be complemented with some novel metrics (based on empirically estimated recall values), which guide developers in the ontology refinement process. Also,

our future work will tackle the exploration of various ways to increase the obtained precision and recall as well as further experiments with more human evaluators.

ACKNOWLEDGMENTS. Amal Zouaq is funded by a postdoctoral fellowship from the Fonds Québécois de la Recherche sur la Nature et les Technologies (FQRNT). This research presented in the paper is in part also supported by the Natural Sciences and Engineering Research Council of Canada through the Discovery Grants – Individual program and the Alberta Innovates – Technology Futures through the New Faculty Award program.

8 REFERENCES

1. Alani, H., Brewster, C. and Shadbolt, N. (2006). Ranking Ontologies with AKTiveRank. In the 5th Int. Semantic Web Conference (ISWC), pp. 5-9, Springer-Verlag.
2. Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. Comput. Linguist. 34, 4 555-596.
3. Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M. and Etzioni, O. (2007). Open information extraction from the Web. In 20th Int. Joint Conf. on AI .pp. 2670-2676, ACM.
4. Barker, K., Agashe, B., Chaw, S. Y., Fan, J., Glass, M., Hobbs, J., Hovy, E., Israel, D., Kim, D.S., Mulkar, R., Patwardhan, S., Porter, B., Tecuci, D., and Yeh, P. (2007). Learning by reading: A prototype system, performance baseline and lessons learned. In the 22nd Nat. Conf. on Artificial Intelligence, pp. 280-286, AAAI Press.
5. Borgatti, S.P. and Everett, M.G. (2006). A Graph-theoretic perspective on centrality, Social Networks, 28(4): 466-484.
6. Bos, J. (2009). Applying automated deduction to natural language understanding. J.Applied Logic 7(1): 100-112.
7. Brewster, C.A. (2008). Mind the gap: bridging from text to ontological knowledge, Ph.D. Thesis, University of Sheffield.
8. Brin, S. & Page, L. (1998). The anatomy of a large-scale hyper-textual web search engine, Stanford University.
9. Buitelaar, P. and Cimiano, P. (Eds.) (2008). Ontology Learning and Population: Bridging the Gap between Text and Knowledge, IOS Press.
10. Cañas, A. J., Carff, R., Hill, G., Carvalho, M., Arguedas, M., Eskridge, T. C., Lott, J., Carvajal R. (2005). Concept Maps: Integrating Knowledge and Information Visualization, in Knowledge and Information Visualization: Searching for Synergies, Springer-Verlag.
11. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr. E. R. and Mitchell, T. M. (2010). Toward an Architecture for Never-Ending Language Learning, Proc. of the 24th Conf. on Artificial Intelligence (AAAI 2010).
12. Cimiano, P. and Völker, J. (2005). Text2Onto. NLDB 2005, pp. 227-238, Springer.
13. Coursey, K. and Mihalcea R. (2009). Topic Identification Using Wikipedia Graph Centrality, in Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL 2009), pp.117-120, Boulder, Colorado.
14. Crysmann, B., Frank, A., Kiefer, B., Muller, S., Neumann, G., Piskorski, J., Schaefer, U., Siegel, M., Uszkoreit, H., Xu, F., Becker, M. and Krieger, H-U (2002). An integrated architecture for shallow and deep processing. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02).* Association for Computational Linguistics, Morristown, NJ, USA, 441-448.
15. De boer, V., Van Somersen, M. & Wielinga, B.J. (2007). A redundancy-based method for the extraction of relation instances from the Web, International Journal of Human-Computer Studies 65(9):816-831, Academic Press Inc.
16. De Marneffe, M-C, MacCartney, B. and Manning. C.D. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In Proc. of LREC, pp. 449-454, ELRA.
17. Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open information extraction from the web. Commun. 51(12): 68-74, ACM.
18. Fortuna, B., Grobelnik, M., & Mladenic, D. (2006). Semi-automatic Data-driven Ontology Construction System. Proc. of the 9th Int. multi-conference Information Society, pp. 309-318, Springer.
19. Frantzi, K.T. and Ananiadou, S. (1999). The C/NC value domain independent method for multi-word term extraction, Journal of Natural Language Processing 3(6): 145-180.
20. Gordon, J.; Van Durme, B.; and Schubert, L. 2010. Learning from the web: Extracting general world knowledge from noisy text. In WikiAI 2010.
21. Hatala, M., Gašević, D., Siadaty, M., Jovanović, J., Torniai, C. (2009) Can Educators Develop Ontologies Using Ontology Extraction Tools: an End User Study, In Proc. of the 4th European Conference on Technology-enhanced Learning, pp. 140-153, Springer-Verlag.
22. Hearst, M. (1992). Automatic Acquisition of Hyponyms from LargeText Corpora. *Proc. of the Fourteenth International Conference on Computational Linguistics*,pp.539-545, Nantes.
23. Hendler, J. (2003). On beyond ontology, keynote talk, Second International Semantic Web Conference, 2003.
24. Hoser, B., Hotho, A., Jaschke, R., Schmitz, C. and Stumme, G. (2006). Semantic network analysis of ontologies. In Y. Sure and J. Domingue, editors, The Semantic Web: Research and Applications, volume 4011 of LNAI, pages 514–529,

Heidelberg, June 2006..
25. Jia, Y., Hoberock, J., Garland, M., and Hart, J. (2008). On the Visualization of Social and other Scale-Free Networks. IEEE Transactions on Visualization and Computer Graphics 14, 6 (Nov. 2008), 1285-1292.
26. Jiang, X. and Tan, A.-H. (2010), CRCTOL: A semantic-based domain ontology learning system. Journal of the American Society for Information Science and Technology, 61: 150–168.
27. JUNG (2010). Last retrieved from http://jung.sourceforge.net/
28. Kim, D. and Porter, B. (2009). Kleo: A Bootstrapping Learning-by-reading System, AAAI Spring Symposium on Learning by Reading and Learning to Read, AAAI Press.
29. Kim, D., Barker, K. and Porter, B. (2009). Knowledge integration across multiple texts, The Fifth International Conference on Knowledge Capture, pp. 49-56, ACM.
30. Krippendorff, Klaus. (2004). Content Analysis: An Introduction to Its Methodology, second edition, Chapter 11. Sage, Thousand Oaks, CA.
31. Kleanthous, S. and Dimitrova, V. (2008). Modeling Semantic Relationships and Centrality to Facilitate Community Knowledge Sharing. In Proc. of Conf. on AH, pp. 123-132,.
32. Klein, D. and Manning, C.D. (2003). Accurate Unlexicalized Parsing. Proc. of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.
33. Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment, Journal of the ACM 46(5): 604-632, ACM.
34. Lin, T., Etzioni, O., and Fogarty, J. (2009). Identifying interesting assertions from the web. In Proc. of the 18th Conf. on information and Knowledge Mgnt, pp. 1787-1790, ACM.
35. Maedche, A. and Volz, R. (2001). The ontology extraction maintenance framework Text-To-Onto, in Proc. of the Wshp on Integrating Data Mining and Knowledge Management.
36. Mika, P. (2005). Ontologies Are Us: A Unified Model of Social Networks and Semantics. International Semantic Web Conference pp. 522-536, Springer.
37. Navigli, R., Lapata, M.: An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation. IEEE Trans. Pattern Anal. Mach. Intell. 32(4): 678-692 (2010)
38. Navigli, R., Velardi, R.: Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. Computational Linguistics 30(2): 151-179 (2004)
39. Özgür, A., Vu T., Erkan, G. and Radev, D. R. (2008). Identifying gene-disease associations using centrality on a literature mined gene-interaction network, Bioinformatics 24(13):277-285
40. Pantel, P. and Pennacchiotti, M. (2008). Automatically Harvesting and Ontologizing Se-mantic Relations. Ontology Learning and Population: Bridging the Gap between Text and Knowledge, pp. 171-198, IOS Press.
41. Pantel, P. and Lin, D. (2002). Discovering Word Senses from Text. In Proceedings of SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 613-619. ACM.
42. Patel, C., Supekar, K., Lee, Y. and Park, E. (2003). Ontokhoj: A semantic web portal for ontology searching, ranking, and classification. In Proc. of the 5th ACM Int. Workshop on Web Information and Data Management, pp. 58–61, ACM.
43. Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5):515–523.
44. SCORM (2010). Last retrieved on 02/04/2010 from http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/SCORM%20Resources
45. Schafer, U. (2007). Integrating Deep and Shallow Natural Language Processing Components. PhD Thesis, Saarland University.
46. Soderland, S. and Mandhani, B. (2007). Moving from Textual Relations to Ontologized Relations, Proceedings of the AAAI Spring Symposium on Machine Reading.
47. Toutanova, K., Klein, D., Manning, C.D. & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network, In Proc. of HLT-NAACL, pp. 252-259.
48. Veksler, V. D., Grintsvayg, A., Lindsey, R., and Gray, W. D. (2007). A proxy for all your semantic needs. 29th Annual Meeting of the Cognitive Science Society, CogSci.
49. Völker, J., Haase, P., and Hitzler, P. 2008. Learning Expressive Ontologies. In Proc. of the Conf. on ontology Learning and Population: Bridging the Gap between Text and Knowledge, pp. 45-69, IOS Press.
50. Wu, F. & Weld, D.S. (2010). Open Information Extraction using Wikipedia. Proc. of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 118-127, ACL.
51. Xie, Z. (2005). Centrality measures in text mining: prediction of noun phrases that appear in abstracts. In Proceedings of the ACL Student Research Workshop, ACL, 103-108.
52. Xu, F., Krieger, H. U. (2003). Integrating shallow and deep NLP for information extraction. RANLP 2003.
53. Zouaq, A., Gasevic, D. & Hatala, M. (2011). Ontologizing Concept Maps using Graph Theory, In Proceedings of the ACM 26th Symposium On Applied Computing, Semantic Web and Applications (To appear).
54. Zouaq, A., Gagnon, M. & Ozell, B. (2010). Semantic Analysis using Dependency-based Grammars and Upper-Level Ontologies, International Journal of Computational Linguistics and Applications, 1(1-2): 85-101, Bahri Publications.

55. Zouaq, A. and Nkambou, R. (2009). Evaluating the Generation of Domain Ontologies in the Knowledge Puzzle Project. IEEE Trans. on Kdge and Data Eng., 21(11): 1559-1572.
56. Zouaq, A. (2008). An Ontological Engineering Approach for the Acquisition and Exploitation of Knowledge in Texts, PhD Thesis, University of Montreal (in French).