Automatic Ontology Identification for Reuse

Mirco Speretta and Susan Gauch
Electrical Engineering and Computer Science
University of Kansas
{mirco, sgauch}@ittc.ku.edu

Abstract

The increasing interest in the Semantic Web is producing a growing number of publicly available domain ontologies. These ontologies are a rich source of information that could be very helpful during the process of engineering other domain ontologies. We present an automatic technique that, given a set of Web documents, selects appropriate domain ontologies from a collection of pre-existing ontologies. We empirically compare an ontology match score that is based on statistical techniques with simple keyword matching algorithms. The algorithms were tested on a set of 183 publicly available ontologies and documents representing ten different domains. Our algorithm was able to select the correct domain ontology as the top ranked ontology 8 out of 10 times.

1. Introduction

The increasing popularity of the Semantic Web has produced a proliferation of ontologies, attracting the interest of many researchers to develop libraries. Ding and Fensel [3] describe the benefits of organizing and reusing available ontologies into libraries. A library of ontologies should provide users with the possibility of re-using, maintaining, adapting and versioning ontologies.

Despite the steady growth, the most common method for building ontologies is still based on manual effort. Ontology engineering employs a variety of different approaches to ontology construction and they are usually based on best practice guidelines. The importance of providing tools to users during the process of constructing ontologies is widely recognized, as shown by the development of projects such as portals and systems for searching, reusing, and distributing ontologies ([4] and [6]).

Rather than starting from scratch for each domain, some projects are investigating reusing existing ontologies for even further efficiency improvements. Although there may be some modifications required, as more and more ontologies become available, it is increasingly likely that third party ontologies might exist that could be used unchanged or, changed at the least, to bootstrap the ontology creation process. Maedche et al. [6] describe in detail the challenges of building systems that reuse ontologies.

The goal of our study is to introduce an automatic technique that can help to identify existing ontologies that would be good candidates for reuse. By automatically exploiting content extracted from sets of Web pages, we employ automatic techniques similar to those employed in ontology learning. On the other hand, rather than building ontologies from scratch by defining taxonomies and building structures, we focus on selecting from existing ontologies whose domain is related to the topic of a given collection of documents in order to bootstrap the ontology learning process.

Our dataset is based on ontologies that we downloaded from publicly available online libraries. For each ontology, we considered only the list of included concepts. No properties or other relationships within the ontologies were taken in consideration in the scope of this study. We assumed that the tokens included in the concepts are a representation of the domain described by the ontology. Statistical weighting techniques were applied to identify the most representative tokens.

2. Background

Velardi et al. [9] gave a comprehensive overview of the state-of-the-art approaches for constructing taxonomies. They also introduced a new semi-automatic technique for creating domain taxonomies. Lately many approaches for searching and reusing ontologies have been proposed. Alani et al. [1] developed AKTiveRank, a prototype system for searching ontologies. For Sabou et al. [8], ontology
selection is the process of identifying an ontology or parts of ontologies that satisfy some given criteria. Jones and Alani [5] also proposed an automatic technique for ranking ontologies.

Our study extends previous ontology reuse work by employing automatic techniques to identify candidate ontologies. We use a set of documents to define the information need and then investigate a variety of approaches to automatically define the topic coverage of the documents and the ontologies. Finally, we extend previous work on ontology ranking by investigating different matching algorithms across a variety of domains.

3. Preliminary Study

Ontology learning and reuse are relatively new topics. The first challenge for this study was to collect the data to be used in the tests. We had to create a collection of ontologies and collect a set of representative Web pages for some of those ontologies to be used in our experiments.

Ontologies were collected from publicly available online libraries. They were downloaded from the following 3 libraries: Protégé1, OntoSelect2, and SchemaWeb3. Table I lists the libraries that we used along with some statistics.

Table 1: List of libraries, number of downloaded ontologies, and number of ontologies included in the collection.

<table>
<thead>
<tr>
<th>Library</th>
<th>Downloaded</th>
<th>Correctly Parsed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protégé</td>
<td>56</td>
<td>80 %</td>
</tr>
<tr>
<td>OntoSelect</td>
<td>1,134</td>
<td>34 %</td>
</tr>
<tr>
<td>SemanticWeb</td>
<td>229</td>
<td>30 %</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1587</strong></td>
<td><strong>31 %</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ontologies Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicates and</td>
</tr>
<tr>
<td>Ontologies with</td>
</tr>
<tr>
<td>more than 50 %</td>
</tr>
<tr>
<td>overlap</td>
</tr>
<tr>
<td>No Natural Languages</td>
</tr>
<tr>
<td><strong>Total</strong></td>
</tr>
<tr>
<td><strong>183</strong></td>
</tr>
</tbody>
</table>

We removed duplicates, i.e., those ontologies with more than 50% overlapping concepts with another ontology as identified using concept names. We also removed ontologies that had no natural language content. This created the final set of 183 ontologies used in our experiments. The number of concepts for the ontologies in this set varied from 13 to 2,622, with an average of 230.

The next step was to select 10 domain ontologies to act as our test set. For each of these, we needed to manually collect a set of Web pages related to the domain to act as examples of the topic coverage. We manually selected a representative set of ontologies that varied in both the number of concepts and the domains represented. Table II shows the list of the domains of the selected ontologies along with some statistics that provide details on the concepts in the 10 selected ontologies.

Table 2. Description of ontologies and sets of documents used for testing

<table>
<thead>
<tr>
<th>ID</th>
<th>Domain</th>
<th>Number of Concepts</th>
<th>Avg Number Tokens per Concept</th>
<th>Total Unique Tokens per set of docs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Music</td>
<td>35</td>
<td>1.3</td>
<td>2935</td>
</tr>
<tr>
<td>2</td>
<td>CS Conference</td>
<td>74</td>
<td>1.7</td>
<td>1493</td>
</tr>
<tr>
<td>3</td>
<td>Beer</td>
<td>51</td>
<td>1.3</td>
<td>3549</td>
</tr>
<tr>
<td>4</td>
<td>Wine</td>
<td>80</td>
<td>2.0</td>
<td>1513</td>
</tr>
<tr>
<td>5</td>
<td>Travel</td>
<td>34</td>
<td>1.4</td>
<td>2419</td>
</tr>
<tr>
<td>6</td>
<td>Linguistic</td>
<td>116</td>
<td>1.8</td>
<td>6226</td>
</tr>
<tr>
<td>7</td>
<td>Pizza</td>
<td>101</td>
<td>2.1</td>
<td>3221</td>
</tr>
<tr>
<td>8</td>
<td>Copyright</td>
<td>113</td>
<td>1.3</td>
<td>3512</td>
</tr>
<tr>
<td>9</td>
<td>Medicine</td>
<td>2622</td>
<td>1.8</td>
<td>1958</td>
</tr>
<tr>
<td>10</td>
<td>Astronomy</td>
<td>952</td>
<td>1.7</td>
<td>2507</td>
</tr>
</tbody>
</table>

For each of the 10 test ontologies, we downloaded 20 Web pages from the associated ODP4 category for a total of 200 documents. All pages were then processed to remove HTML tags, punctuation, stop words, and the surviving words were reduced to their stems using the Porter stemmer [7].

4. Approach

Our data set includes 183 ontologies along with a set of 20 sample documents for each of the 10 test ontologies. Thus, the domain of each set of documents matched at least one ontology. Given a set of sample documents, the goal of our study was to automatically find the ontology that we used to find the ODP category (and thus the sample documents). We based our initial approaches on matching all words from the set of documents and all the words included in the concept descriptions in the ontologies. All the algorithms that we implemented and analyzed have two inputs: 1) the set of documents from a particular category and 2) the set of all 183 ontologies. The output is a ranked list of ontologies where the ontology with the higher score is judged more likely to be the correct match for the set of documents.

We conducted two sets of experiments. The first, which was used as a baseline, applied a string matching

---

1 http://protege.cim3.net/cgi-in/wiki.pl?ProtegeOntologiesLibrary
2 http://olp.dfki.de/OntoSelect/w/index.php?mode=home
3 http://schemaweb.info/
4 http://dmoz.org/
technique, and the second was based on statistical weighting techniques.

The statistical weighting technique is based on the ontology match \( om(DS,O_i) \) score that represents the quality of the match between each ontology and the given set of documents. The ontology match was calculated as the sum of term match values for all the preprocessed tokens extracted from the concept names in the ontology.

\[
om(DS,O_i) = \sum_{t \in DS \cap O_i} tm_{DS,O_i}(t) \tag{1}
\]

where

- \( DS \) is the document set being processed;
- \( O \) is the ontology set, unless otherwise stated and \( O_i \in O \) is the current ontology being matched;
- \( tm_{DS,O_i}(t) \) is the term match value for term \( t \) in \( DS \cap O_i \);

The term match value is defined as

\[
tm_{DS,O_i}(t) = w_{DS}(t) \times w_O(t) \tag{2}
\]

where

- \( w_{DS}(t) \) is the weight of \( t \) in a given set of documents \( DS \);
- \( w_O(t) \) is the weight of \( t \) in the ontology \( O \);

\[
w_{DS}(t) = tf_{DS}(t) \times idf(t) \tag{3}
\]

where

- \( tf_{DS}(t) \) is the frequency of \( t \) in \( DS \);
- \( idf(t) \) is the inverse document frequency;

\[
idf(t) = \log \left( \frac{N_{ADS}}{n_{ADS}(t)} \right) \tag{4}
\]

where

- \( N_{ADS} \) is the number of all documents from all sets;
- \( n_{ADS}(t) \) is the number of documents in the given set which contains \( t \).

\[
w_O(t) = tf_O(t) \times iof(t) \tag{5}
\]

where

- \( tf_O(t) \) is the frequency of \( t \) in the given ontology \( O \);
- \( iof(t) \) is the inverse ontology frequency of \( t \) in the given ontology \( O \);

\[
ifo(t) = \log \left( \frac{|C_O|}{n(t)} \right) \tag{6}
\]

\( |C_O| \) is the number of concepts in the given ontology \( O \);
\( n(t) \) is the number of ontologies with term \( t \).

Tests were run for the following variations:

1. the terms extracted from the documents and ontologies were matched with and without stemming;
2. the percentage of the top weighted terms extracted from the set of documents used to match against the ontology terms was varied from 10% to 100% in steps of 10%.
3. the ontology match score normalized with the following 3 techniques: magnitude of the ontology, number of concepts in the ontology and total number of words.

5. Results

The goal of the first experiment was to identify the correct ontology by looking at the percentages of matched concepts using tokens from the given set of documents. The average rank for the correct ontologies was 35.9. The use of stems gave very similar results: the average rank for the correct ontologies was 40.6, a little higher than using tokens.

For the following experiments we applied statistical weighting techniques. Tokens and stems were considered separately. For each, we compared the percentage of tokens from the set of documents used for matching and the normalization algorithm. The output of the algorithm is a ranked list of ontologies. Ontologies closer to the top position are the most likely to represent the domain of the given set of documents. Thus a lower value of average rank represents a better outcome of the algorithm.

The data in Figure 1 shows the average rank of the correct ontologies. Normalizing by the number of concepts or tokens gave very similar results: the correct ontology for each set of documents was consistently ranked within the top 4 with an average of 1.7.

We then ran the same set of experiments using stems instead of tokens for the terms representing both
document sets and concepts. Figure 2 shows that the best performing algorithm places the correct ontology at an average rank of 1.2. We observe that normalizing the number of tokens performs slightly better than normalizing by the number of concepts, but the difference is very small when we use stems. This is explained by the fact that the number of concepts is more closely correlated with the number of stems than tokens. Either using tokens or stems we can observe that, as long as a normalization algorithm is applied, the results are independent from the percentage of tokens that were used from the set of documents.

7. Conclusions and Future Work

In this study, we introduced a novel approach for reusing ontologies automatically. We showed that, in our experimental setting, our algorithm was able to rank the correct ontology within the top 2 for all 10 test domains. The application of our technique has two main benefits. First, ontology engineers can be provided with a new tool to speed up the process of building ontologies. Second, it is not a computationally expensive technique, thus it could be used in application were the time response is critical (e.g., interactive applications deployed over the Internet).

We applied statistical techniques to find candidate domain ontologies for a given set of documents. Thus, we only tackled the problem of finding an appropriate set of concepts. Further investigation is needed to measure the quality of the matches between the top ranked ontologies and the given set of documents. We plan to instantiate the candidate ontologies with the given set of documents. Then, we will be able to study the number of concepts semantically related to the set of documents.

Belkin [2] states that recommendation systems should help users to fill the gap of knowledge that they are seeking. We think that the same principles should be applied in the context of ontology selection. User’s subjectivity, especially when dealing with knowledge representation and taxonomies should be taken in consideration when trying to reuse ontologies. Thus, a completely automatic approach might not be the best solution for all users.

10. References


