TermExtractor: a Web Application to Learn the Common Terminology of Interest Groups and Research Communities

Francesco Sclano, Paola Velardi
Dipartimento di Informatica, via Salaria 113 Roma
velardi@di.uniroma1.it, francesco_sclano@yahoo.it

Abstract

We implemented a high-performing technique to automatically extract from the available documents in a given domain the shared terminology of a web community. This technique has been successfully experimented and submitted for large-scale evaluation in the domain of enterprise interoperability, by the member of the INTEROP network of excellence. In order to make the technique available to any web community in any domain, we developed a web application that allows users to i) acquire (incrementally or in a single step) a terminology in any domain, by submitting documents of variable length and format, and ii) validate on-line the obtained results. The system also supports collaborative evaluation by a group of experts. The web application has been widely tested in several domains by many international institutions that volunteered for this task.

1. Introduction

In (Navigli and Velardi, 2004) we presented a technique, named OntoLearn, to automatically learn a domain ontology from the documents shared by the members of a web community. This technique is based on three learning steps, each followed by manual validation: terminology extraction, glossary extraction, and finally, ontology enrichment. The OntoLearn methodology has been enhanced (Navigli and Velardi, 2005) and experimented in real settings (Velardi et al. 2007). Recently, we started to develop web applications to make freely available each of the steps of the OntoLearn methodology. This paper describes TermExtractor, a tool to extract the terminology “shared” among the members of a web community, through the analysis of the documents they exchange. Defining a domain lexicon is in fact the first step of an ontology building process.

The contributions of the paper, with respect to published work, are the following:

1. We summarize the terminology extraction algorithm, on which we provided already a description in (Navigli and Velardi, 2002), with the main intent of highlighting progress wrt previously reported work;
2. We summarize the features and options of the TermExtractor web application;
3. We provide an evaluation of the web tool performed by a world-wide group of TermExtractor users, who volunteered to perform the task.

---

1 This paper intentionally focuses on the web application and evaluation more than on term extraction algorithms. The reader is invited to read the referred papers for additional details.
The main novelty is the evaluation procedure, considerably more wide-coverage than in available literature, where the standard methodology to evaluate terminology extraction systems is based on three judges with adjudication, on a single domain. Second, TermExtractor is able to learn the terminology from a corpus of documents, not just a single document. Finally, to the best of our knowledge, the TermExtractor web application is the most sophisticated freely available terminology-extraction tool to date.

2. The term extraction algorithms

As many terminology extraction systems (Wermter and Hahn, 2005) (Bourigault and Jacquemin, 1999) (Park et al., 2002), in TermExtractor the identification of relevant terms is based on two steps: first, a linguistic processor is used to parse text and extract typical terminological structures, like compounds (enterprise model), adjective-noun (local network) and noun-preposition-noun (board of directors). Then, the (usually large) list of terminological candidates is purged according to various filters.

Figure 1 shows the main processing phases of the term extraction module.

TermExtractor combines the best reported terminology extraction techniques, including those introduced in (Navigli and Velardi, 2002). We summarize here the filters used to exclude non-terminological multi-word strings from the list of syntactically plausible candidates:

**Domain Pertinence**: let $D_i$ be the domain of interest (represented by a set of relevant documents) and let $D_1, D_2, D_{i-1}, D_{i+1}, \ldots, D_N$ be sets of documents (or terminologies) in other domains, e.g. medicine, economy, politics, etc. The Domain Relevance of a term $t$ wrt a domain $D_i$ is measured as:

\[
DR_{D_i}(t) = \frac{\hat{P}(t/D_i)}{\max_j \hat{P}(t/D_j)} = \frac{\max_j \text{freq}(t,D_j)}{\max_j \text{freq}(t,D_i)}
\]

The Domain Pertinence is high if a term is frequent in the domain of interest and much less frequent in the other domains used for contrast. A similar filtering criterion is used also in (Park et al. 2002).

**Domain Consensus**: this measure, which is a novel measure that we introduced in (Navigli and Velardi, 2002), simulates the consensus that a term must gain in a community before being considered a relevant domain term. The Domain Consensus is an entropy-related measure, computed as:

\[
\hat{P} \text{ is the Expected Value of the probability } P.
\]
where \( d_k \) is a document in \( D_i \) and \( \text{norm-freq} \) is a normalized term frequency. The domain consensus is then normalized for each term in the \([0,1]\) interval. The consensus is high if a term has an even probability distribution across the documents chosen to represent the domain.

**Lexical Cohesion**: this measure evaluates the degree of cohesion among the words that compose a terminological string \( t \). This measure is described in (Park et al. 2002) and proved to be more effective than other measures of cohesion in literature. Let \( |t| = n \) be the length of \( t \) in number of words. The lexical Cohesion is measured as:

\[
LC_{D_i}(t) = \frac{n}{\sum_{w_j \in D_i} \text{freq}(w_j, D_i)} \log(\text{freq}(t, D_i))
\]

where \( w_j \) are the words composing the term \( t \). The cohesion is high if the words composing the term are more frequently found within the term than alone in texts.

**Structural Relevance**: if a term is highlighted in a document, e.g. it appears in the title or paragraph title, or if it is in bold or underlined etc., then the measure of its frequency, used in formulas (1-3) is increased by a factor \( k \) (user-adjustable).

**Miscellaneous**: a set of heuristics are used to remove from terms generic modifiers (e.g. \textit{large} knowledge base), to detect mispellings\(^3\) (using the WordNet on-line dictionary), to distinguish terminology from proper nouns, to extract single-word terminology, to detect acronyms, etc. We omit these details for sake of space, but the reader can access the web application and related documentation on http://lcl.di.uniroma1.it/termextractor.

The final weight of term is a linear combination of the three main filters:

\[
w(t, D_i) = \alpha \cdot DR + \beta \cdot DC + \gamma \cdot LC
\]

where the coefficients are user-adjustable, but the default is: \( \alpha = \beta = \gamma = \frac{1}{3} \).

### 3. The web application

The TermExtraction web application has a pipeline architecture composed of 6 main phases, shown in Figure 2:

1. **Set Termextractor options**: in this phase the user can set several options or accept the default;
2. **Upload documents**: the user can upload documents in almost any format, one by one or in zipped archives;
3. **Convert documents**: documents are converted in txt format, heuristics are used to correct conversion errors (especially originated by pdf files);
4. **Term Extraction**: in this phase the terminology is extracted and filtered (see previous section);
5. **Terminology Validation**: in this phase a partner or a team of partners validate the terminology;
6. **Save-download Terminology**: in this phase the terminology is saved or downloaded in txt, xml or xcl format.

\(^3\) Errors are often generated when converting documents in pdf or ps format in txt, see next Section.
At the end of phase 2 the user is disconnected\(^4\), to allow for intensive data processing and to handle multiple users. At the end of extraction process (phases 3 and 4) the user receives an e-mail and is addressed to the validation page (phase 5). After the validation, the user can download the terminology, or he can save it on the TermExtractor server for further extension or validation (phase 6). The terminology is stored on the server for a limited time period (two weeks). We now provide some detail on each phase.

Figure 2. TermExtractor Pipeline Architecture

In phase 1 the user is asked to set several options, or to accept the default options. For sake of brevity, we mention here only the most relevant settings\(^5\):

Select-deselect contrastive corpora: contrastive corpora are used to compute the Domain Relevance, through the formula (1). Examples of domains used for contrastive analysis are medicine, computer networks, etc. However, if the user domain of interest is, e.g. wireless networks, he may want to capture some more generic term in the area of computer networks. In this case, he can access the “Terminology” option and exclude Computer Networks from the list of domains used to compute the denominator of formula (1).

Set minimum and maximum length of terms: with this option the user can set the minimum and maximum length of multi-word terms to be extracted. Single-word terms are mostly (but not exclusively) extracted a posteriori, selecting the most frequent singleton components of multi-word terms.

Adjust the coefficients of the weight formula: here the user can tune the formula (4) by adjusting the three coefficients.

Other available options include: whether or not detecting proper nouns, whether to acquire a brand-new terminology, or enrich (and upload) an already existing one, selection of layout features to be considered in Structural Relevance analysis, etc.

Figure 3 shows one of the option windows, relative to layout analysis.

In phase 2 the user can upload a set of documents that he considers relevant to model the domain under analysis. The effectiveness of TermExtractor filters depends on statistical significance, therefore in general, larger corpora obtain better results. Figure 4 shows the document-uploading interface. The user can upload up to 20 different documents, or as many documents he wants, compressed in a zipped archive. All main document formats are processed, as listed in the central box in Figure 4. It is also possible to specify the url of a web page. Once the documents have been uploaded, the user is disconnected from the application. When document processing is completed (phases 3 and 4 of Figure 2), he receives an e-mail pointing to the webpage where the evaluation can take place (phase 5). Single users can perform the evaluation, however, in real settings, a domain terminology must be evaluated and accepted consensually by a team of domain specialists. This is, for example, the evaluation procedure followed in the INTEROP EC project, as detailed in section 4.

\(^4\) a “demo version” is available in which the user can upload a single document and obtain the result immediately.

\(^5\) the interested reader can inspect the application and the available options on http://lcl.di.uniroma1.it
Consensus building, which is particularly relevant when the objective is to learn the “domain lexicon” of a distributed community, is supported by a dedicated validation interface. A coordinator of the validation process must qualify himself, select the validation team, and establish the start and end date of the validation, as shown in Figure 5.
Then, each partner of the validation team can inspect the terminology and accept or reject terms. The interface allows ordering extracted terms according to the global weight (formula 4), or to each of four indicators: Domain Relevance, Consensus, Cohesion and Frequency. Evaluators are also allowed to propose new terms and insert them in the list. This is shown in Figure 6. At the end of the validation period (but also during the validation), the coordinator can inspect the final result, as shown in Figure 7. The vote cumulated by each term is shown in the rightmost column.

In the final phase 6, the user can download the validated terminology in one of several formats (txt, xml, xls etc.). He can also temporarily save the terminology on the TermExtractor server, for further extensions.

4. Evaluation

One of the novel aspects of the work presented in this paper is evaluation. Unlike many existing systems for terminology extraction (see bibliography), TermExtractor has been validated in the large by web communities and individual users on different domains.

As previously mentioned, the term extraction algorithm (described in section 2) was first used within the EC INTEROP project6 to extract a domain lexicon in the area of enterprise interoperability research. The interoperability lexicon was validated by the members of the Network of Excellence7, and the results of the evaluation are shown in Table I. The extraction of the interoperability lexicon was the first step of the OntoLearn knowledge acquisition methodology, described in (Velardi et al. 2007), that eventually brought to the creation of a domain taxonomy8. Table I shows that the activity of term validation was greatly participated

---

6 INTEROP is a network of excellence on enterprise interoperability research. The web site is http://www.interop-noe.org
7 In INTEROP, the term extractor program has been used off-line. Only team validation was supported by a web application.
8 The INTEROP taxonomy is browsable in http://lcl.di.uniroma1.it/tav and http://interop-noe.org/backoffice/km/domains
by the NoE members, with around 2500 expressed votes. Overall, the precision of the system\(^9\) was around 60%, but perhaps the most relevant result has been the speed-up of the lexicon creation process supported by TermExtractor. In emergent web communities, like the INTEROP NoE, often the domain of interest is not well-assessed, therefore the task of manually identifying the relevant domain terms by a team of specialists is difficult and time-consuming. In INTEROP, the community was composed by researchers belonging to rather heterogeneous fields, namely: ontology and knowledge representation, enterprise modeling, architectures and platforms. The “enterprise interoperability” domain was initially weakly defined, therefore capturing the common, relevant concepts was one of the first targets of the NoE. TermExtractor fostered the identification of many emergent domain concepts, and furthermore it provided a valuable support to consensus building, which is a key issue during the definition of a domain terminology.

---

\(^9\) note that the evaluation of TermExtractor from the INTEROP NoE took place on November 2005. After that date, many improvements in the algorithm have been added.
The team of evaluators includes a restricted group of INTEROP partners and was enriched with web users showing some interest in the application and volunteering to participate in the evaluation. The results shown in Table II are rather encouraging. In Table II, INTEROP members are highlighted (but the documents used for test where different from those used to extract the INTEROP lexicon), the other are external users who volunteered to perform the evaluation. Users 2 and 5 are private companies, the other are research institutions. In the table, the precision is automatically computed by the system, depending upon the number of rejected terms, while the global judgment is specified by each evaluator, selecting in a dedicated window from 5 possible judgments ranging from very good to bad.

5. Related research

Most available on-line term extraction services\(^\text{10}\) use very simple extraction algorithms: typically, they extract every word and every phrase up to a certain number of words in length that occurs at least a minimum number of times in a source text file and that do not start or end with a stop word. The document to be analyzed can either be a web page, specified through its *url*, or a plain text submitted by the user.

---

\(^{10}\) e.g. Topicalizer [http://www.topicalizer.com/] and Textalyser [http://textalyser.net/]
To the best of our knowledge, the only terminology learning application with comparable complexity of the extraction algorithms is the IBM Glossex system (Park et al. 2002), (Kozakov et al. 2004). Similarly to TermExtractor, Glossex filters terminological candidates using lexical cohesion and a measure of domain relevance. Glossex has also some additional useful heuristics, like aggregation of lexical variants for a term (e.g. compounding variants like fog lamps and foglamps, or inflectional variants like rewinding and rewound). On the other side, Glossex analyses one document at a time, therefore it is unable to identify popular domain terms with an even probability distribution across the documents of a collection (like we do with Domain Consensus measure). Unfortunately, a performance comparison based on the analysis of single documents is not possible since the Glossex tool is not freely available.

We finally stress that in virtually all papers on terminology extraction mentioned so far the validation is conducted manually by three judges (usually the authors themselves). This is not comparable with the large-scale evaluation of TermExtractor conducted within web communities like INTEROP, and by several external institutions around the world, on different domains and competences.

References

BOURIGAULT D. AND JACQUEMIN C.: Term Extraction+Term Clustering: an integrated platform for computer-aided terminology, in Proc. of EACL , 1999


NAVIGLI R., P. VELARDI "Structural Semantic Interconnections: a knowledge-based approach to word sense disambiguation" Special Issue-Syntactic and Structural Pattern Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 27, n. 7, 2005


<table>
<thead>
<tr>
<th>Company/Organization</th>
<th>Vote</th>
<th>Terminology description (as Indicated by user)</th>
<th># of documents</th>
<th># of terms before validation</th>
<th># of terms after validation</th>
<th>Precision&lt;sup&gt;13&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Pavel Terminology Tutorial - Translation Bureau - Government of Canada</td>
<td>Very Good</td>
<td>A chapter from a Collective Agreement</td>
<td>1</td>
<td>39</td>
<td>37</td>
<td>0.948</td>
</tr>
<tr>
<td>Stockholm University</td>
<td>Very Good</td>
<td>English anatomy/medical terms.</td>
<td>7680</td>
<td>4457</td>
<td>4432</td>
<td>0.994</td>
</tr>
<tr>
<td>K-Tech</td>
<td>Very Good</td>
<td>Terminology about design pattern from 5 different books</td>
<td>5</td>
<td>549</td>
<td>534</td>
<td>0.973</td>
</tr>
<tr>
<td>IASI - Institute of Systems Analysis and Computer Science. Rome, Italy. (INTEROP partner)</td>
<td>Good</td>
<td>Ontology alignment, Ontology mapping, Ontology matching, model transformation</td>
<td>58</td>
<td>157</td>
<td>145</td>
<td>0.923</td>
</tr>
<tr>
<td>School of Information Technology and Engineering, University of Ottawa, Canada</td>
<td>Very Good</td>
<td>Terminology extracted from the Nuclear-Free Canada program.</td>
<td>7</td>
<td>575</td>
<td>513</td>
<td>0.892</td>
</tr>
<tr>
<td>Computational Linguistics Research, Damascus, MD, US.</td>
<td>Very Good</td>
<td>Terminology developed from Senseval-3 papers.</td>
<td>19</td>
<td>58</td>
<td>49</td>
<td>0.844</td>
</tr>
<tr>
<td>Università Politecnica delle Marche, Ancona, Italy. (INTEROP partner)</td>
<td>Good</td>
<td>Security Protocols</td>
<td>40</td>
<td>158</td>
<td>128</td>
<td>0.810</td>
</tr>
<tr>
<td>Jena University Language and Information Engineering (JULIE) Lab, Germany</td>
<td>Good</td>
<td>Immunology (It’s a broad branch of biomedical science that covers the study of all aspects of the immune system in all organisms).</td>
<td>1</td>
<td>35</td>
<td>25</td>
<td>0.714</td>
</tr>
<tr>
<td>Federal University of Ceara</td>
<td>Very Good</td>
<td>Embodied cognitive science</td>
<td>5</td>
<td>131</td>
<td>85</td>
<td>0.648</td>
</tr>
<tr>
<td>Università Politecnica delle Marche, Ancona, Italy. (INTEROP partner)</td>
<td>Good</td>
<td>Semantic Similarity</td>
<td>16</td>
<td>80</td>
<td>49</td>
<td>0.612</td>
</tr>
<tr>
<td>CIMOSA</td>
<td>Good</td>
<td>Standardisation ISO TC 184, SC5 and related activities</td>
<td>18</td>
<td>102</td>
<td>62</td>
<td>0.607</td>
</tr>
<tr>
<td>Universidad Politecnica de Valencia, Spain. (INTEROP partner)</td>
<td>Good</td>
<td>Collaborative Networks from the point of view of Enterprise Modelling</td>
<td>63</td>
<td>115</td>
<td>60</td>
<td>0.521</td>
</tr>
<tr>
<td>Waikato University, New Zealand</td>
<td>Good</td>
<td>Semantic Web and Ontologies</td>
<td>1</td>
<td>42</td>
<td>35</td>
<td>0.833</td>
</tr>
<tr>
<td>Department for Computational Linguistics, University of Potsdam</td>
<td>Very Good</td>
<td><em>Thinking in C++</em> Vol.1 - Bruce Eckels</td>
<td>32</td>
<td>237</td>
<td>231</td>
<td>0.974</td>
</tr>
</tbody>
</table>

Table II- Evaluation of the TermExtractor web application by several web users

<sup>11</sup> Volunteers accepted to publish the name of their institution  
<sup>12</sup> Possible votes were: Very Good, Good, Fair, Poor and Bad.  
<sup>13</sup> Precision = (Number of terms before validation) / (Number of terms after validation)