TermExtractor: a Web Application to Learn the Shared Terminology of Emergent Web Communities

FRANCESCO SCLANO

University of Roma "La Sapienza"
francesco_sclano@yahoo.it

PAOLA VELARDI

University of Roma "La Sapienza"
velardi@di.uniroma1.it

Abstract. In the Semantic Web era, many techniques have been proposed to capture the explicit knowledge of a virtual community, and represent this knowledge in a structured form often referred to as *domain ontology*. One of the first steps of the ontology-building task is to collect a vocabulary of domain relevant terms. We designed a high-performing technique to automatically extract the shared terminology from available documents in a given domain. 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 the members of *any* web community, we developed a web application that allows users to acquire (incrementally or in a single step) a terminology in any domain, by submitting documents of variable length and format, and 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 recent years, a growing number of communities and networked enterprises started to access and interoperate through the Internet. Modelling these communities and their information needs is important for several web applications, like topic-driven crawlers [1], web services [2], recommender systems [3], etc.

One of the first steps to model the knowledge domain of a virtual community is to collect a vocabulary of domain-relevant terms, constituting *the linguistic surface manifestation* of domain concepts. Several methods to automatically extract technical terms from domain-specific document ware-houses have been described in the literature e.g, [4,5,6,7,8]. Typically, approaches to automatic term extraction make use of linguistic processors (part of speech tagging, phrase chunking) to extract terminological candidates. Terminological entries are then filtered from the candidate list using statistical and machine learning methods.

In [9,10] we presented a novel technique to filter domain terminology from candidates multi-word expressions that, a part from the efficacy of the specific statistical and linguistic filters used, had three main distinguishing features:

To perform a contrastive analysis of terms, wrt a collection of corpora in other domains;

To simulate with a specific statistical indicator the level of consensus a term must gain before it is actually accepted within a community;

To enhance statistical filters with structural analysis[a] of the texts from which a candidate term is extracted (position of the term in the document structure).

In the mentioned papers, the system, named TermExtractor, was evaluated in several domains through *domain experts voting with adjudication*[b], a commonly adopted evaluation procedure. Very recently, TermExtractor has been submitted for large-scale evaluation in the domain of enterprise interoperability, by the members of a network of excellence, the INTEROP NoE[c] [11].

Given the good performance of our tool, and its potential utility to support the modelling of web communities, we implemented a web application, which is extensively described in the rest of this paper. The web application will be offered as a service to other communities through the European Association being constituted as a continuation of the INTEROP project, the so-called Virtual Laboratory on Enterprise Interoperability.

In Section 2 we briefly describe the term extraction algorithms and its recent extensions. Section 3 and 4 are dedicated to a detailed description of the TermExtractor Web Application. Section 5 describes an evaluation experiment in which we involved several international institutions that volunteered to perform the task, based on their interest for the application. A summary result of the INTEROP evaluation experiment is also provided. Finally, Sections 7 is dedicated to the state of art and future extensions.

2. The term extraction algorithms

As many terminology extraction systems, 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 sequences (board of directors). Then, the (usually large) list of terminological candidates is purged according to various filters.

In TermExtractor, we use the following filters:

Domain Pertinence: let Di be the domain of interest (represented by a set of relevant documents) and let D_1,D_2, D_{i-1},D_{i+1},..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 Di is measured as[d]:

$$(1)\quad DR_{D_i}(t) = \frac{\hat{P}(t/D_i)}{\max_j\left(\hat{P}(t/Dj)\right)} = \frac{freq(t,D_i)}{\max_j\left(freq(t,Dj)\right)}$$

The domain pertinence is high is a term is frequent in the domain of interest and much less frequent in the other domains used for contrast. A similar measure is used also in [8].

---

[a] This feature has been recently added and is not discussed in []

[b] In case of inter-annotator disagreement, either a majority voting is adopted, or the annotators are requested to produce a uniform vote after a discussion.

[c] http://www.interop-noe.org

[d] $\hat{P}$ is the expected value of the probability P.

Domain Consensus: this measure, which is novel with respect to terminology extraction algorithms in literature, simulates the *consensus* that a term must gain in a community before being considered a stable domain term. The domain consensus is an entropy-related measure, computed as:

(2)

$$DR_{D_i}(t) = -\sum_{d_k \in D_i} \hat{P}(t/d_k)\log(\hat{P}(t/d_k)) = -\sum_{d_k \in D_i} norm-freq(t,d_k)\log(norm-freq(t,d_k))$$

where d_k is a document in D_i and 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 of the domain.

Lexical Cohesion: this measure evaluates the degree of cohesion among the words that compose a terminological string t. This measure has been introduced in [8] 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:

(3) $$LC_{D_i}(t) = \frac{n \cdot freq(t,D_i) \cdot \log(freq(t,D_i)}{\sum_{w_j} freq(w_j,D_i)}$$

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 an integer k (user-adjustable).

Miscellaneous: a set of heuristics are used to remove from terms generic modifiers (e.g. *large* knowledge base), to detect mispellings (using the WordNet on-line dictionary), to distinguish terms from proper nouns, etc. We omit these details for sake of space.

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

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

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

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

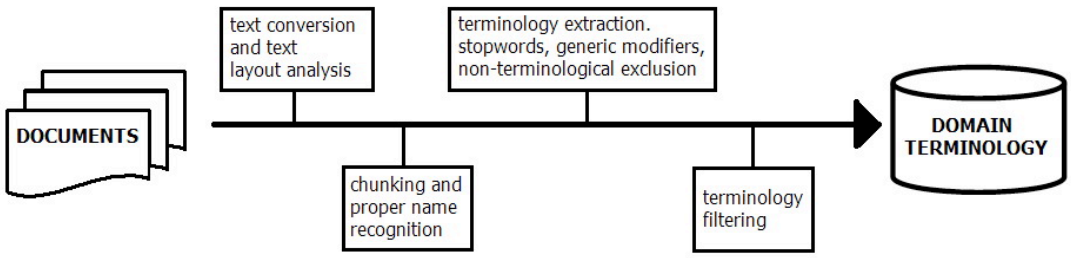

Figure 1. Terminology extraction phases.

Additional details are presented in the next section.

3. The term extraction web application: architecture

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

1. Set Termextractor options: in this phasethe user can set several options or leave the default
2. Upload documents: the user can upload documents in many formats or zipped archives
3. Convert documents: documents in almost any format are converted in txt format
4. Term Extraction: in this phase the terminology is extracted and filtered
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 or xml format.

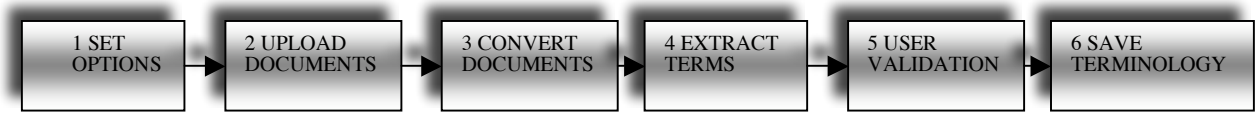

Figure 2. TermExtractor Pipeline Architecture

At the end of phase 2 the user is disconnected[e], 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 in *xml* or text format, 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).

TermExtractor is implemented in *java enterprise* and is composed by the following main modules:

- **termextractor**, a java application which is the "hart" of the system
- **txtconverter**,a library to convert in txt format the documents uploaded by the user in various formats
- **runshandler**, a library to manage asynchronous threads[f] (in TermExtractor an asynchronous thread is a terminology extraction process)
- **termextractorweb**, the web application available on http://lcl2.di.uniroma1.it/termextractor

TermExtractor uses the following programs:

- **JBoss**, is an open source java enterprise-based application server
- **Tomcat**, a web server that supports servlets and JSPs
- **MySQL**, a multithreaded, multi-user, SQL Database Management System (DBMS)
- **WordNet**, a semantic lexicon for the English language
- **TreeTagger** is a tool for annotating text with part-of-speech and lemma information

and some additional java libraries.

[e] a demo mode is available in which the user can upload a single document and see the result immediately

[f] Processes that proceed independently of each other. Using the client-server model, the server handles many asynchronous requests from its many clients. The client is often able to proceed with other work.

4. Description of the interface: features and options

This section provides a brief overview of the phases listed in previous section.

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:

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.

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

Figure 3 shows one of the option windows.

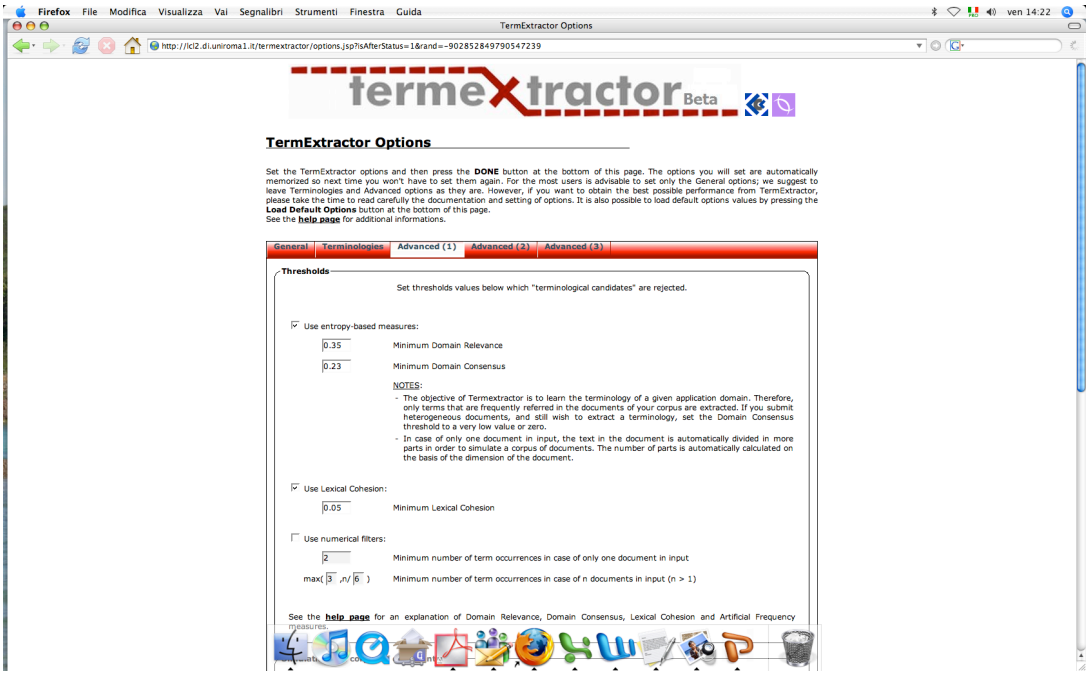

Figure 3 . Setting term extraction options

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, which type of textual highlights

(bold, italic, underlined..) to consider in Structural Relevance analysis, the adjustment of the k parameter (see section 2), etc.

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.

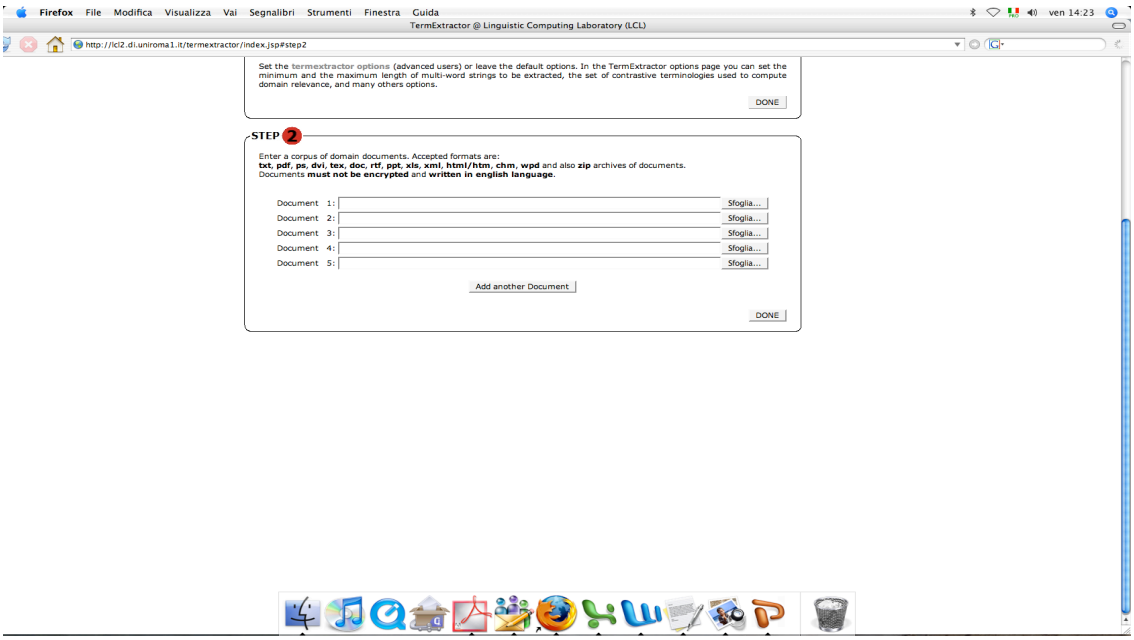

Figure 4. Document uploading

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 web page 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 procedure followed in the INTEROP project, as detailed in section 5.

Consensus building 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. For each term the global cumulated vote is shown in the rightmost column.

In the final **phase 6**, the user can download the validated terminology in one of several formats (text, xml, etc.). he can also temporarily save the terminology on the TermExtractor server.

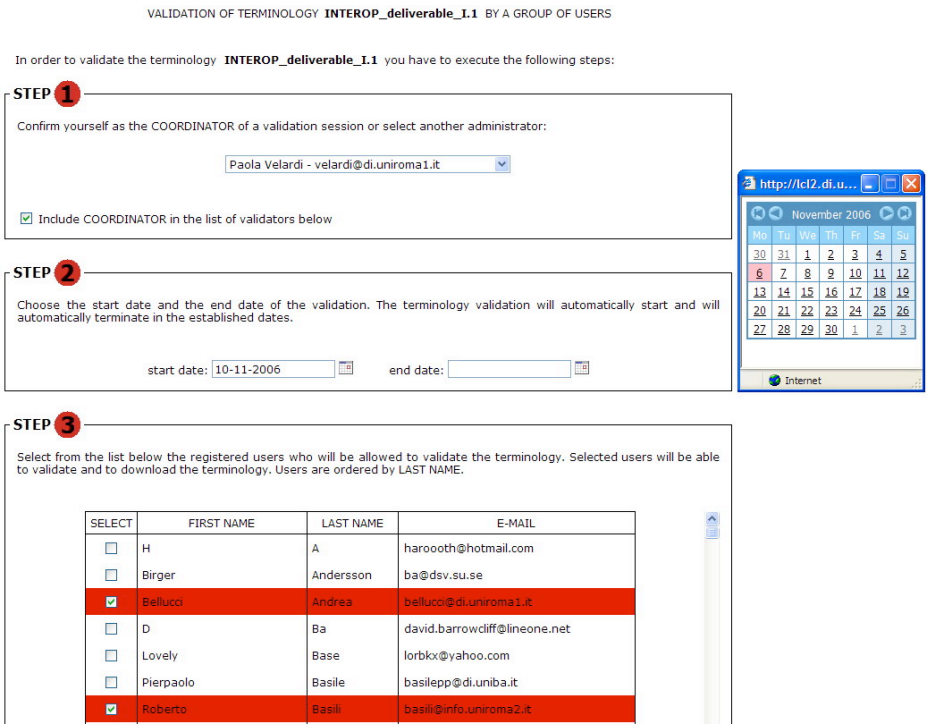

Figure 5. Launching a team validation

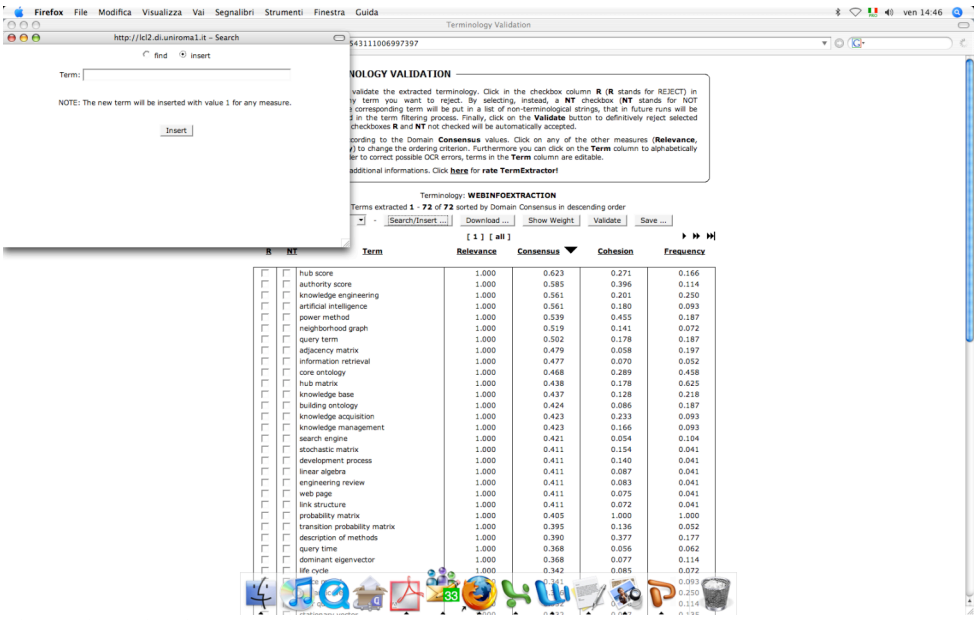

Figure 6. Validation by a single partner

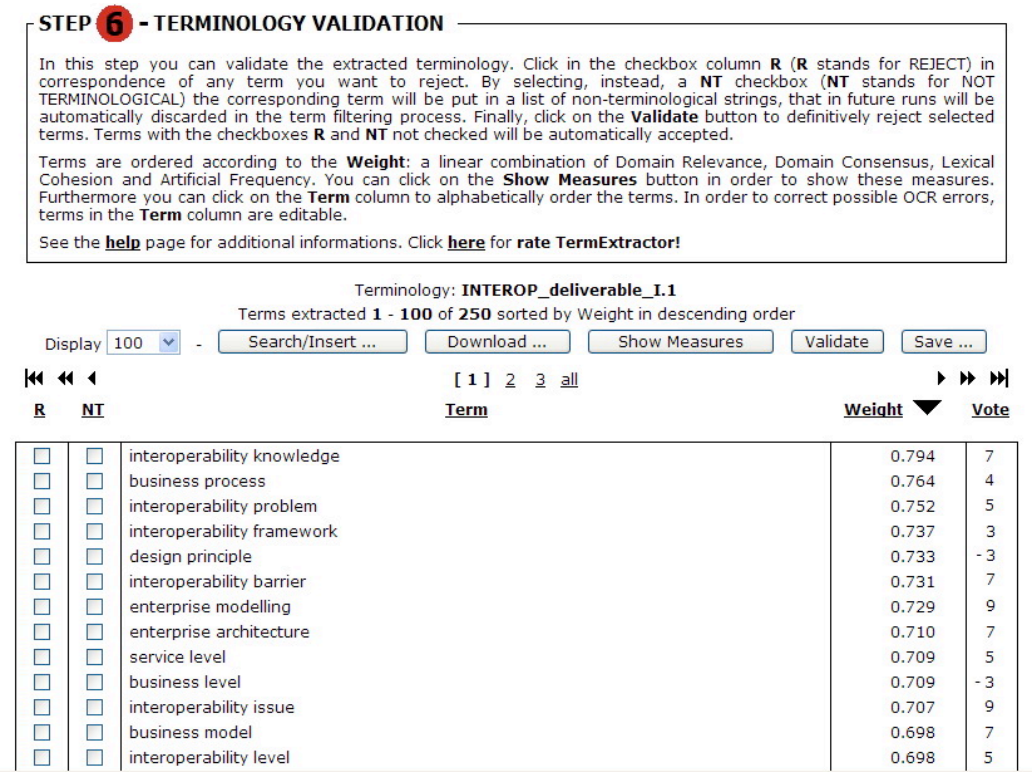

Figure 7. Inspecting the results of a team validation

5.Evaluation of TermExtractor

One of the relevant aspects of the work presented of this paper is evaluation. Unlike many existing systems for terminology extraction (see section 6), TermExtractor has been validated in the large by web communities and individual users around the world.

Initially, the term extraction program (section 2) has been used within the INTEROP project to extract a domain lexicon in the area of enterprise interoperability research. The extracted lexicon has been collectively validated by the entire network of Excellence[g], and the results are shown in Table 1. The interoperability lexicon has been the first result of a knowledge acquisition process, described in [11], that eventually brought to the creation of a domain taxonomy[h].

The table shows that the activity of term validation was greatly participated by the NoE members, with around 2500 expressed votes. Overall, the precision of the system[i] was around 60%, but perhaps

[g] In INTEROP, the term extractor program has been used as a stand-alone. Only team validation was supported by a web application.

[h] The INTEROP taxonomy is browsable in http://lcl.di.uniromai.it/tav and http://interop-noe.org/backoffice/km/domains

[i] note that the evaluation of TermExtractor from the INTEROP NoE took place one year ago. After that date, many improvements in the algorithm have been added.

the most relevant result has been the speed-up of the lexicon creation process allowed by TermExtractor.

In emergent web communities, like the INTEROP NoE, often the domain of interest is not well-assessed, therefore the task of identifying the relevant domain terms by a team of specialists is not an easy one.

In INTEROP, the community was composed by researchers belonging to rather heterogeneous fields: 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 foreseen tasks 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.

n. of extracted terms	1902
Total voting partners	35
Total expressed votes	2453
Total different terms with a negative vote	783 (41%)
Survived terms	1120

Table 1. Summary of INTEROP collaborative terminology validation (November 2005)

After the validation of the term extraction methodology in INTEROP, it was decided to create a web application, to make the extraction service available outside the Network of Excellence, and to demonstrate the generality of the approach.

The result of the INTEROP collective validation was used to add further improvements to the system, which was finally uploaded and made available on the web in October 2006.

To evaluate the precision and the user-friendliness of the application, we asked several institutions around the world to test the system on different domains, and provide a global judgment.

The team of evaluators includes a restricted team of INTEROP partners and is being enriched with institutions showing some interest in the application[j] and volunteering to participate in the evaluation. This evaluation is still in progress, but the preliminary results, shown in Appendix, are rather encouraging. In the Appendix table, Partners 1,3,4,5 and 6 are INTEROP members, the other are external and volunteered to perform the evaluation driven by their interest. 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.

6. Related research and concluding remarks

Most available on-line term extraction services[k] 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 does not start or end with a stop word. The

[j] These users have been selected detecting multiple submissions of tasks to the TermExtractor application.

[k] e.g. Topicalizer http://www.topicalizer.com/ and Textalyser http://textalyser.net/

document to be analysed can either be a web page, specified trough its *url*, or a plain text submitted by the user. Figure 8 shows the result of submitting to the *Topicalizer* system the abstract of this article.

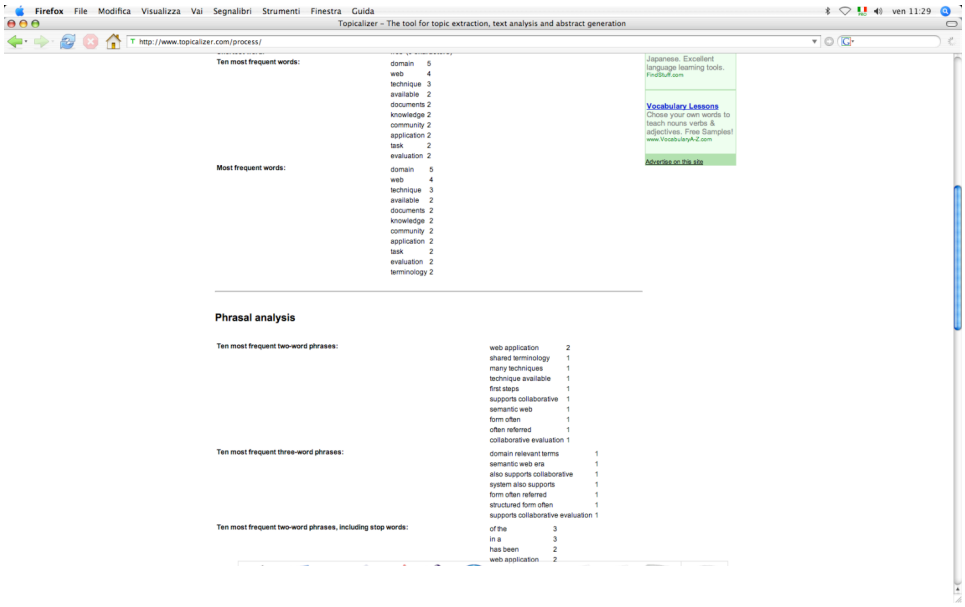

Figure 8. A screen dump of Topicalizer system

To the best of our knowledge, the only terminology learning application with comparable complexity of the extraction algorithms is the IBM Glossex system ([7,8]).

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. compunding 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 (the Consensus measure). Unfortunately, a performance comparison based on the analysis of single documents is not possible since the Glossex tool is not freely available.

In [7,8], as in virtually all papers on terminology extraction [4,5,6], the validation is conducted manually by three judges (usually the authors themselves). This is not comparable with the large-scale evaluation of TermExtractor (that, by the way, is still in progress), conducted within web communities like INTEROP, and by several external institutions around the world, with different domains and competences.

In conclusion, we believe that i) accuracy of the extraction process, ii) speed-up of terminology acquisition versus manual methodologies (as proposed e.g. in the METHONTOLOGY framework [12]), and iii) support to team validation, make of TermExtractor one of the best available terminology extraction applications, as also confirmed by the growing community of users and by their positive comments, some of which revealed also very useful to add further improvements to the system.

8 References

1. Menczer F., Pant G. and Srinivasan P. : Topic-Driven Crawlers: machine learning issues, in http://citeseer.ist.psu.edu/menczer02topicdriven.html

2. Fan J. and Kambhampati S. : A Snapshot of Public Web Services, in ACM SIGMOD Record archive Volume 34 , Issue 1 (March 2005)

3. Yan Zheng Wei, Luc Moreau, Nicholas R. Jennings: A market-based approach to recommender systems, in ACM Transactions on Information Systems (TOIS), Volume 23 Issue 3, July 2005

4. Wermter J. and Hahn U.: Finding New terminology in Very large Corpora, in Proc. of K-CAP'05, October 2-5, 2005, Banff, Alberta, Canada

5. Bourigault D. and Jacquemin C.: Term Extraction+Term Clustering: an integrated platform for computer-aided terminology, in Proc. of EACL , 1999

6. Collier N., Nobata C. and Tsujii J. : Automatic acquisition and classification of terminology using a tagged corpus in the molecular biology domain, Terminology, 7(2). 239-257, 2002

7. L. Kozakov, Y. Park, T. Fin, Y. Drissi, Y. Doganata, and T. Cofino "Glossary extraction and utilization in the information search and delivery system for IBM Technical Support", IBM System Journal, Volume 43, Number 3, 2004

8. Y. Park, R. J. Byrd, B. Boguraev "Automatic glossary extraction: beyond terminology identification" International Conference On Computational Linguistics , Proceedings of the 19th international conference on Computational linguistics – Taipei, Taiwan, 2002

9. Navigli R. and Velardi, P. (2004). "Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites". Computational Linguistics. vol. 50 (2).

10. Navigli R. Velardi, P., Cucchiarelli A. and Neri F: (2004). Quantitative and Qualitative Evaluation of the OntoLearn Ontology Learning System. 20th COLING 2004, Geneva, August 2004

11. P. Velardi A. Cucchiarelli and M. Pètit "A Taxonomy Learning Method and its Application to Characterize a Scientific Web Community" IEEE Transactions on Knowledge and Data Engineering, in press, 2007

12. M. Fernández, A. Gómez-Pérez, N. Juristo "METHONTOLOGY: From Ontological Art Towards Ontological Engineering" Spring Symposium Series. Stanford. PP: 33-40, 1997

Appendix. Preliminary results of web-wide evaluation of TermExtractor

Organization[l]	Vote[m]	Terminology description	Number of submitted documents	Number of terms before validation	Number of terms after validation	Precision[n]
1. IASI	Good	Ontology alignment, Ontology mapping, Ontology matching, model transformation	58	157	145	0.923
2. CL Research	Very Good	Word Sense Disambiguation	19	58	49	0.844
3. DIIGA – University of Ancona	Good	Security Protocols	40	158	128	0.810
4. DIIGA - University of Ancona	Good	Semantic Similarity	16	80	49	0.612
5. CIMOSA	Good	Standardisation ISO TC 184 SC5 and related activities	18	102	62	0.607
6. Universidad Politecnica de Valencia	Good	Collaborative Networks from the point of view of Enterprise Modelling	63	115	60	0.521
7. Waikato University	Good	Semantic Web and Ontologies	1	42	35	0.833
8. Department for Computational Linguistics, University of Potsdam.	Very Good	"Thinking in C++" Vol.1 - Bruce Eckels	32	237	231	0.974
Average Precision						0.765

[l] Volunteers accepted to publish the name of their institution

[m] Possible votes were Very Good, Good, Fair, Poor and Bad.

[n] Precision = (Number of terms before validation) / (Number of terms after validation)