TBXTools: A Free, Fast and Flexible Tool for Automatic Terminology Extraction

Antoni Oliver
Universitat Oberta de Catalunya
aoliverg@uoc.edu

Mercè Vàzquez
Universitat Oberta de Catalunya
mvazquezga@uoc.edu

Abstract

The manual identification of terminology from specialized corpora is a complex task that needs to be addressed by flexible tools, in order to facilitate the construction of multilingual terminologies which are the main resources for computer-assisted translation tools, machine translation or ontologies. The automatic terminology extraction tools developed so far either use a proprietary code or an open source code, that is limited to certain software functionalities. To automatically extract terms from specialized corpora for different purposes such as constructing dictionaries, thesauruses or translation memories, we need open source tools to easily integrate new functionalities to improve term selection. This paper presents TBXTools, a free automatic terminology extraction tool that implements linguistic and statistical methods for multiword term extraction. The tool allows the users to easily identify multiword terms from specialized corpora and also, if needed, translation candidates from parallel corpora. In this paper we present the main features of TBXTools along with evaluation results for term extraction, both using statistical and linguistic methodology, for several corpora.

1 Introduction

Automatic terminology extraction (ATE) is a relevant natural language processing task involving terminology which has been used to identify domain-relevant terms applying computational methods (Oliver et al., 2007a; Foo, 2012).

Automatic term extraction is a relevant task that can be useful for a wide range of tasks, such as ontology learning, machine translation, computer-assisted translation, thesaurus construction, classification, indexing, information retrieval, and also text mining and text summarisation (Heid and McNaught, 1991; Frantzi and Ananiadou, 1996; Vu et al., 2008).

The automatic terminology extraction tools developed in recent years allow easier manual term extraction from a specialized corpus, which is a long, tedious and repetitive task that has the risk of being unsystematic and subjective, very costly in economic terms and limited by the current available information. However, existing tools should be improved in order to get more consistent terminology and greater productivity (Gornostay, 2010).

In the last few years, several term extraction tools have been developed, but most of them are language-dependent: French and English –Fastr (Jacquemin, 1999) and Acabit (Daille, 2003); Portuguese –Extracterm (Costa et al., 2004) and ExATOlp (Lopes et al., 2009); Spanish-Basque –Elexbi (Hernaiz et al., 2006); Spanish-German –Autoterm (Haller, 2008); Arabic (Boulaknadel et al., 2008); Slovene and English –Luiz (Vintar, 2010); English and Italian –KX (Pianta and Tonelli, 2010); or English and German (Gojun et al., 2012).

Some tools are adapted to a specialized domain: TermExtractor (Sclano and Velardi, 2007), TerMine (Ananiadou et al., 2009) or BioYaTeA (Golik et al., 2013), for example. Specific tools have been developed to extract corpus-specific lexical items comparing technical and non-technical corpus: TermoStat (Drouin, 2003). And other tools are based on under-resourced language –TWSC (Pinnis et al., 2012)–, or use semantic and contextual information –Yate (Vivaldi and Rodríguez, 2001).

Furthermore, there was TermSuite, which was developed during the European project TTC (*Terminology Extraction, Translation Tools and Com-*

parable Corpora). This project focused on the automatic or semi-automatic acquisition of aligned bilingual terminologies for computer-assisted translation and machine translation. To this end, automatic terminology extraction is part of the process of identifying terminologies from comparable corpora (Blancafort et al., 2010).

This paper presents TBXTools, a free automatic term extraction tool which allows multiword terms from specialized corpora to be identified easily, combining statistical and linguistic methods.

This paper is structured as follows: in the next section we present the TBXTools implementation and statistical and linguistic methods, as well as the automatic finding of translation equivalents. The experimental settings are described in detail in section 3. The paper concludes with some final remarks and ideas for future work.

2 TBXTools

2.1 Description

TBXTools is a Python class that implements a set of methods for ATE along with other utilities related to terminology management. This tool has a free software licence and can be downloaded from SourceForge[1]. TBXTools is an evolution of previous tools developed by the authors (Oliver and Vàzquez, 2007; Oliver et al., 2007b).The tool is still under development but it already implements a set of methods that permit the following functionalities:

- Statistical term extraction using n-grams and stop words and allowing some normalizations: capital letter normalization, morphological normalization and nested candidate detection.

- Linguistic term extraction using morphosyntactic pattern and a tagged corpus. Any external tagger and a connection with a server running Freeling (Padró and Stanilovsky, 2012) are implemented. The tool uses an easy formalism for the expression of patterns, allowing the use of regular expressions and lemmatization of some of the components, if required.

- Detection of translation candidates in parallel corpora, using a statistical strategy.

- Automatic learning of morphological patterns from a list of reference terms.

Nowadays TBXTools does not have a user interface, but it will be developed in the future. At present the extraction is done by means of simple Python scripts calling the TBXTools class. In this paper we will see the code of some of these scripts. Several examples of scripts can be found in the TBXTools distribution.

2.2 Statistical Terminology Extraction

The statistical strategy for terminology extraction is based on the calculation of n-grams, that is, the combination of n words appearing in the corpus. After this calculation, filtering with stop words is performed, eliminating all the candidates beginning or ending with a word from a list. Some normalizations, such as case normalization, nesting detection and morphological normalization, can be performed. Here we can see a complete code for terminology extraction:

```
from TBXTools import *
e=TBXTools()
e.load_sl_corpus("corpus.txt")
e.load_stop_l1("stop-eng.txt")
e.set_nmin(2)
e.set_nmax(3)
e.statistical_term_extraction()
e.case_normalization()
e.nesting_detection()
e.load_morphopatterns("morpho-eng.txt")
e.morpho_normalization()
e.save_term_candidates("candidates.txt")
```

The code, as can be seen, is very simple. First of all, we import TBXTools and create a TBXTools object, called e in the example. This code calculates the term candidates from the corpus in the $corpus.txt$ file using the stop words in the $stop.txt$ file. Afterwards, we fix the minimum n to 2 and the maximum to 3, in order to calculate bigrams and trigrams term candidates. The next step in the code performs the statistical term extraction. After that, the following normalizations are implemented:

- Case normalization: it tries to collapse the same term appearing with a different case: for example, “interest rate”, “Interest Rate” and “INTEREST RATE” into “interest rate”.

- Nesting detection: sometimes shorter term candidates are not terms in and of themselves, but are part of a longer term. For example, the bigram term candidate “national central” is a part of the trigram term candidate “national central bank”.

[1] http://sourceforge.net/projects/tbxtools/

- Morphological normalization: it tries to collapse several forms at the same time into a single form, for example, to collapse the plural term candidate "economic policies" into "economic policy". To perform this normalization, a simple set of morphological patterns is used. After all these normalizations, the term candidates are saved into the text file *candidates.txt*. The candidates are stored in descending frequency order and the value of frequency is also stored, as in the following example:

```
53 euro banknotes
51 central bank
47 payment institution
23 payment instrument
```

2.3 Linguistic Terminology Extraction

To perform linguistic terminology extraction we need a POS-tagged corpus. The tagging can be performed with any tagger offering lemma and POS tags. TBXTools can be easily used with Freeling. In the following example we will perform linguistic extraction from a tagged corpus (*ct.txt*) using a set of patterns (p) and storing the term candidates into the file *candidates.txt*. The Python script would look like this:

```
from TBXTools import *
e=TBXTools()
e.load_tagged_corpus("ct.txt")
e.load_ling_termextract_patterns("p.txt")
e.ling_term_extract()
e.save_term_candidates("candidates.txt")
```

If our tagged corpus uses the Penn Treebank POS tags, the patterns should be expressed with these same tags, for example NN NN or JJ NN. If we want to use the lemma instead of the word form in a pattern, we use square brackets, as in NN [NN.*]. Note that in this pattern we have also used regular expressions to make it more general. The formalism also allows for the inclusion of the lemmas and word forms in the patterns, as in [N.*] /of/ [N.*], where the lemma *of* is used.

TBXTools is able to calculate the translation equivalent for a given term using a parallel corpus. If the given term appears several times in the corpus, TBXTools can use simple statistical calculations to try to select the translation equivalent in the target language. In the following code we can observe how this task can be performed:

```
from TBXTools import *
import codecs
e=TBXTools()
e.load_tabtxt_corpus("corpus.txt")
e.load_stop_l2("stop.txt")
...
tr=e.get_statistical_translation_
candidate(t, candidates=5)
print(t,tr)
...
```

With this code we load a parallel corpus and a list of stop words for the target language. Then we calculate the translation equivalent (tr) from the term (t) and ask to return 5 candidates. The output would as follows:

```
payment institution entidad de pago:
servicios de pago:dinero electrónico:
entidad de crédito:Estado miembro:
```

In this example we want to find the translation of "payment institution" and we get 5 candidates in Spanish. In this case the first one is the correct one ("entidad de pago").

3 Experimental Settings

3.1 Resources

We performed some experiments on terminology extraction using controlled corpora, that is, we knew in advance which terms are in these corpora. We used a subset of 1,000 segments from the ECB (European Central Bank) corpus and EMEA (European Medicines Agency documents corpus) corpus (Tiedemann, 2012) in English.

A manual selection of terms in these corpus subsets was performed. Terms in the corpus were manually annotated and those in plural form were lemmatized. This annotation task was performed independently by two terminologists, and those cases with no agreement were discussed and a common solution adopted. Having these annotated corpora, we extracted a list of all terms and their frequencies. Two different lists were extracted for each corpus: a list containing the terms as they appeared in the corpus (in plural or lemma form), and another list containing only the lemmatized terms. These lists of extracted terms from the manually annotated corpora were used to evaluate the extraction results.

3.2 Methodology

In our experiments we performed and evaluated 3 different tasks for both corpus subsets:

- Statistical terminology extraction for English
- Linguistic terminology extraction for English
- Automatic extraction of translation equivalents into Spanish

In all these experiments we used TBXTools. The programs used have been described in section 2.

3.3 Evaluation and Results

Since we have a list of all terms appearing in both corpus subsets, evaluation of the automatic terminology extraction experiments could be done automatically. We have evaluated precision for different values of frequency. TBXTools has a method that, given a set of translation candidates, a list of terms and a value of frequency, calculates the precision and recall values. Here we can see a piece of code for the evaluation task:

```
...
e.load_evaluationterms("ref_terms.txt")
(p,r)=extractor.eval_prec_recall_byfreq(5)
...
```

This code returns the value of precision (p) and recall (r) for all candidates with a frequency of 5 or higher.

The task of automatic extraction of translation equivalents has been evaluated manually by a terminologist.

Statistical Approach

In tables 1 to 4 we can see the evaluation results for the statistical approach. We have presented figures of precision ($P.$) and reacall ($R.$) for bigrams and trigrams and for the ECB and EMEA subsets of 1,000 segments. As we can observe in all results, for high values of frequency we get very few term candidates and the values of precision are not significant, as recall is too low.

In Table 1 we can observe the results for the statistical approach using the subset of the ECB corpus. The total number of candidates for bigram word forms are 720, and for bigram lemmata 696. If we focus on figures for frequency equal to 2, we get 280 candidates with a precision of 43.21% for word forms and 274 candidates with a precision of 27.37% for lemmata. This significant difference between these two values (15.84 points) indicates that the simple approach to lemmatization based on morphological normalization using simple morphological patterns is not very accurate.

	Word forms		Lemmata	
Freq	**P.**	**R.**	**P.**	**R.**
50	100.00	0.34	50.00	0.41
20	100.00	2.06	71.43	2.03
10	61.54	5.50	57.14	6.50
5	59.09	13.40	41.27	10.57
2	43.21	41.58	27.37	30.49
1	29.58	73.20	17.10	48.37

Table 1: Results for statitistical approach using ECB corpus for bigrams

In Table 2 we can observe the results for trigrams. The total number of candidates for trigram word forms are 726, and for trigram lemmata 722. As we can see, the precision values for trigrams are worse than for bigrams (for frequency equal to 2, from 43.21% to 18.72% for word forms and from 27.37% to 5.47% for lemmata).

	Word forms		Lemmata	
Freq	**P.**	**R.**	**P.**	**R.**
50	0	0	X	X
20	100.00	1.87	50.00	2.38
10	75.00	2.80	33.33	4.76
5	50.00	9.35	25.00	11.90
2	18.72	35.51	5.47	26.19
1	10.06	68.22	2.08	35.71

Table 2: Results for statistical approach using ECB corpus for trigrams

In tables 3 and 4 the results for the EMEA subcorpus are presented. The total number of candidates for bigram word forms is 432, and for bigram lemmata, 422, whereas for trigrams the total is 367 both for word forms and lemmata. The behaviour here is very similar to that of the ECB corpus, but here the number of bigram and trigram candidates is lower than for the ECB corpus.

Linguistic Approach

In tables 5 to 8 the results for the linguistic approach are presented.

	Word forms		Lemmata	
Freq	**P.**	**R.**	**P.**	**R.**
50	0	0	0	0
20	100.00	1.90	100.00	2.84
10	77.78	8.86	66.67	8.51
5	52.24	22.15	42.42	19.86
2	30.41	70.25	22.50	57.45
1	27.78	75.95	20.38	60.99

Table 3: Results for statistical approach using EMEA corpus for bigrams

	Word forms		Lemmata	
Freq	**P.**	**R.**	**P.**	**R.**
50	0	0	0	0
20	100.00	2.33	100.00	2.38
10	28.57	4.65	14.29	2.38
5	13.89	11.63	8.33	7.14
2	9.70	67.44	6.69	47.62
1	8.45	72.09	5.99	52.38

Table 4: Results for statistical approach using EMEA corpus for trigrams

	Word forms		Lemmata	
Freq	**P.**	**R.**	**P.**	**R.**
20	100.00	0.69	66.67	0.81
10	66.67	2.75	75.00	4.88
5	58.14	8.59	57.50	9.35
2	41.10	33.33	36.48	34.55
1	25.82	67.70	23.26	69.11

Table 5: Results for linguistic approach using ECB corpus for bigrams

	Word forms		Lemmata	
Freq	**P.**	**R.**	**P.**	**R.**
20	100.00	0.93	0	0
10	33.33	0.93	0	0
5	30.77	3.74	15.38	4.76
2	13.95	16.82	6.98	21.43
1	9.36	49.53	3.55	47.62

Table 6: Results for linguistic approach using ECB corpus for trigrams

	Word forms		Lemmata	
Freq	**P.**	**R.**	**P.**	**R.**
20	100.00	2.53	100.00	3.55
10	87.50	8.86	83.33	10.64
5	66.67	21.52	66.00	23.40
2	29.77	81.01	29.12	86.52
1	28.69	84.81	27.97	90.07

Table 7: Results for linguistic approach using EMEA corpus for bigrams

	Word forms		Lemmata	
Freq	**P.**	**R.**	**P.**	**R.**
20	0	0	0	0
10	16.67	2.33	16.67	2.38
5	12.00	6.98	11.11	7.14
2	9.27	67.44	9.74	71.43
1	8.71	72.09	9.43	78.57

Table 8: Results for linguistic approach using EMEA corpus for trigrams

For the extraction of bigrams candidates we have used a set of patterns that have been learnt with TBXTools. This feature uses the tagged corpus and a set of reference terms and returns a list of patterns. This list should be manually revised and modified in order to make the patterns more general.

In Table 7 the results for the linguistic approach using the ECB corpus for bigrams are presented. For frequency equal to 2, a precision of 41.10% for word forms and 36.48% for lemmata is achieved. If we now observe the difference between these values (a difference of 4.62 points instead of the 15.84 points for morphological normalization in the statistical approach), we can conclude that the linguistic approach performs much better in the task of normalizing the terms into their base form.

Automatic Extraction of Translation Equivalents in Parallel Corpora

In this section we present the results for the experiments with automatic extraction of translation equivalents in parallel corpora. The Spanish equivalents selection for the English terms (in lemma form) in ECB and EMEA subcorpora was done by two experts translators. As TBXTools is able to return several translation candidates for each corpora, we assessed if the first candidate was correct (P_1) and if any of the first five candidates were correct (P_5). As the algorithm did not produce Spanish translations for many English terms, we also presented a corrected precision ($P{*}_1$ and $P{*}_5$), taking only into account the English terms for which the algorithm returned some translation candidates. In some cases we failed to find the translation of a term because we searched using the lemma form and the term always appeared in plural in the corpus. Tables 9 and 10 shows the recall values.

Table 9 shows the evaluation results using a parallel corpus consisting of the first 1,000 seg-

	$\mathbf{P}_1$	$\mathbf{P}_5$	$\mathbf{P^*}_1$	$\mathbf{P^*}_5$	$\mathbf{R}_1$	$\mathbf{R}_5$
ECB 2g	12.60%	26.01%	27.93%	57.66%	12.60%	26.01%
ECB 3g	2.78%	12.96%	10%	46.67%	2.78%	12.96%
EMEA 2g	23.40%	43.97%	34.02%	63.92%	23.40%	43.97%
EMEA 3g	2.38%	35.70%	4.00%	60.00%	2.38%	35.71%

Table 9: Results for automatic extraction of translation equivalents for 1,000 segments subcorpora

	$\mathbf{P}_1$	$\mathbf{P}_5$	$\mathbf{P^*}_1$	$\mathbf{P^*}_5$	$\mathbf{R}_1$	$\mathbf{R}_5$
ECB 2g	30.89%	47.15%	46.63%	71.17%	30.89%	47.15%
ECB 3g	11.11%	36.11%	21.05%	68.42%	11.11%	31.48%
EMEA 2g	49.65%	68.79%	56.00%	77.60%	49.65%	62.25%
EMEA 3g	16.67%	52.38%	22.58%	70.97%	16.67%	52.35%

Table 10: Results for automatic extraction of translation equivalents for the full corpora

ments of the corpora (the same subset used for extracting the English term candidates). It is evident that precision for bigrams is much higher than precision for trigrams. This is mainly due to the fact that, in general, frequency for trigram terms is much lower than for bigram terms. This fact becomes less important when we correct the results excluding these terms with no translation candidates.

Table 10 shows the evaluation results using the full corpora for finding the translation candidates. As can be observed, precision and recall values are now much higher, as more English sentences can be found containing the desired term, and therefore there are more Spanish sentences with which to find the translation equivalent.

4 Conclusions and Future Work

This paper has presented a free automatic terminology extraction tool. This tool is written in Python and it can work under any popular operating system. The tool is designed to achieve the following:

- The tool is fast and efficient.
- The tool is flexible, allowing several techniques and normalizations to be used.
- It works in terminal and the user only needs to write simple Python scripts. No Python programming knowledge is required, as scripts are simple and readable. The user can make new scripts by copying and modifying example scripts.
- It is designed to work under Python 2.X and 3.X, without the need for external libraries, avoiding installation problems.

This tool is still under development but it can be used to build monolingual or bilingual terminology glossaries in a fast and efficient way.

In the near future we plan to add the following features:

- Statistical measures for term candidate reordering.
- Improved algorithm for automatic learning of patterns for linguistic terminology extraction.
- Implementation of an algorithm for learning morphological variants of term candidates.
- Development of a simple visual user interface, to make the use of TBXTools even more easy.

In this paper we have also presented the results of the experiments for statistical and linguistic monolingual terminology extraction and for the automatic detection of translation equivalents in parallel corpora.

References

Sophia Ananiadou, Brian Rea, Naoaki Okazaki, Rob Procter, and James Thomas. 2009. Supporting systematic reviews using text mining. *Social Science Computer Review*.

Helena Blancafort, Béatrice Daille, Tatiana Gornostay, Ulrich Heid, Claude Méchoulam, and Serge Sharoff. 2010. TTC: Terminology Extraction, Translation Tools and Comparable Corpora. In European association for lexicography, editor, *14th EURALEX International Congress*, pages 263–268, Leeuwarden/Ljouwert, Netherlands, July.

Siham Boulaknadel, Beatrice Daille, and Driss Aboutajdine. 2008. A multi-word term extraction program for arabic language. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, page 1485–1488, Marràqueix, Marroc. European Language Resources Association.

Rute Costa, Raquel Silva, and Maria Teresa Lino. 2004. Extracterm: an extractor for portuguese language. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC04)*, pages 1–5.

Béatrice Daille. 2003. Conceptual structuring through term variations. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment-Volume 18*, pages 9–16. Association for Computational Linguistics.

Patrick Drouin. 2003. Term extraction using non-technical corpora as a point of leverage. *Terminology*, 9(1):99–115.

Jody Foo. 2012. *Computational Terminology: Exploring Bilingual and Monolingual Term Extraction*. Ph.D. thesis, Linköping University, Linköping, Suècia.

Katerina Frantzi and Sophia Ananiadou. 1996. A hybrid approach to term recognition. In *Proceedings of the International Conference on Natural Language Processing and Industrial Applications (NLP-IA 1996)*, volume 1, page 93–98, Moncton, Canadà.

Anita Gojun, Ulrich Heid, Bernd Weissbach, Carola Loth, and Insa Mingers. 2012. Adapting and evaluating a generic term extraction tool. In *LREC*, pages 651–656.

Wiktoria Golik, Robert Bossy, Zorana Ratkovic, and Nédellec Claire. 2013. Improving term extraction with linguistic analysis in the biomedical domain. In *Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing13), Special Issue of the journal Research in Computing Science*, pages 24–30.

Tatiana Gornostay. 2010. Terminology management in real use. In *Proceedings of the 5th International Conference Applied Linguistics in Science and Education*, pages 25–26.

Johann Haller. 2008. Autoterm: Term candidate extraction for technical documentation (spanish/german). *Tradumàtica: traducció i tecnologies de la informació i la comunicació*, (6).

Ulrich Heid and John McNaught. 1991. EUROTRA-7 study: Feasibility and project definition study on the reusability of lexical and terminological resources in computerised applications. Final report. CEC-DG XIII.

Antton Gurrutxaga Hernaiz, Xavier Saralegi Urizar, Sahats Ugartetxea, and Iñaki Alegría Loinaz. 2006. Elexbi, a basic tool for bilingual term extraction from spanish-basque parallel corpora. In *Atti del XII Congresso Internazionale di Lessicografia: Torino, 6-9 settembre 2006*, pages 159–165.

Christian Jacquemin. 1999. Syntagmatic and paradigmatic representations of term variation. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 341–348. Association for Computational Linguistics.

Lucelene Lopes, Paulo Fernandes, Renata Vieira, and Guilherme Fedrizzi. 2009. Eχatolp–an automatic tool for term extraction from portuguese language corpora. In *Proceedings of the 4th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC), Faculty of Mathematics and Computer Science of Adam Mickiewicz University*, pages 427–431.

Antoni Oliver and Mercè Vàzquez. 2007. A free terminology extraction suite. Londres. Information Management.

Antoni Oliver, Joaquim Moré, and Salvador Climent. 2007a. *Traducció i tecnologies*, volume 116 of *Manuals*. Editorial UOC, Barcelona.

Antoni Oliver, Mercè Vázquez, and Joaquim Moré. 2007b. Linguoc lexterm: una herramienta de extracción automática de terminología gratuita. *Translation Journal*, 11(4).

Lluís Padró and Evgeny Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey, May. ELRA.

Emanuele Pianta and Sara Tonelli. 2010. Kx: A flexible system for keyphrase extraction. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 170–173. Association for Computational Linguistics.

Marcis Pinnis, Nikola Ljubešić, Dan Ştefănescu, Inguna Skadiņa, Marko Tadić, and Tatiana Gornostay. 2012. Term extraction, tagging, and mapping tools for under-resourced languages. In *Proceedings of the 10th Conference on Terminology and Knowledge Engineering (TKE 2012), June*, pages 20–21.

Francesco Sclano and Paola Velardi. 2007. Termextractor: a web application to learn the shared terminology of emergent web communities. In *Enterprise Interoperability II*, pages 287–290. Springer.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1246.

Špela Vintar. 2010. Bilingual term recognition revisited the bag-of-equivalents term alignment approach and its evaluation. *Terminology*, 16(2):141–158.

Jorge Vivaldi and Horacio Rodríguez. 2001. Improving term extraction by combining different techniques. *Terminology*, 7(1):31–48.

Thuy Vu, Ai Ti Aw, and Min Zhang. 2008. Term extraction through unithood and termhood unification. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP 2008)*, volume 1, pages 631–636, Hyderabad, Índia.