Procesamiento del Lenguaje Natural, Revista nº 47 septiembre de 2011, pp 65-73 recibido 03-05-2011 aceptado 25-05-2011

Extracting terminology from Wikipedia

Extracción de terminología a partir de la Wikipedia

Jorge Vivaldi
Applied Linguistic Institute, UPF
Roc Boronat 138, Barcelona, Spain
jorge.vivaldi@upf.edu

Horacio Rodríguez
Software Department, UPC
Jordi Girona Salgado 1-3, Barcelona, Spain
horacio@lsi.upc.edu

Resumen: En este artículo presentamos una aproximación novedosa para obtener la terminología de un dominio utilizando las estructuras de páginas y categorías de Wikipedia de una forma independiente del dominio y de la lengua. La idea es aprovechar el grafo de categorías de Wikipedia a partir de un conjunto de categorías que asociamos con el dominio. Después de obtener las categorías del dominio seleccionado se extraen las páginas correspondientes con ciertas restricciones. El conjunto resultante de páginas y categorías se seleccionan como vocabulario inicial del dominio. Comparamos los resultados obtenidos mediante un modulo de un extractor híbrido, YATE y su equivalente que utiliza la Wikipedia. El resultado muestra que este recurso puede utilizarse para esta tarea. Aplicamos esta aproximación a cuatro dominios (astronomía, química, economía y medicina) y dos idiomas (inglés y castellano).
Palabras clave: extracción de términos, Wikipedia.

Abstract: In this paper we present a new approach for obtaining the terminology of a given domain using the category and page structures of the Wikipedia in a domain and language independent way. The idea is to take profit of category graph of Wikipedia starting with a set of categories that we associate with the domain. After obtaining the full set of categories belonging to the selected domain, the collection of corresponding pages is extracted, using some constraints. The set of titles of recovered pages and categories is selected as initial domain term vocabulary. The system has been evaluated substituting by it the term candidates analyzer module of an state-of-the-art term extractor, YATE. The results show that this resource may be used for this task overcoming some of the limitations of alternative knowledge sources. This approach has been applied to three domains (astronomy, chemistry, economics and medicine) and two languages (English and Spanish).
Keywords: term extraction, term recognition, Wikipedia.

1 Introduction

Although many NLP resources and processors are, or are claimed to be, domain independent, the application of most of such resources or processors to specific NLP tasks uses to be restricted to specific domains and/or genres. As the accuracy of NLP processors degrades heavily when applied in environments (domain, genre) different from which they were built or learned, a process of tuning of the resources and processors to the new environment is usually needed. The basic knowledge sources needed for performing this tuning process are domain restricted corpora and terminological lexicons. A nice example of facing the extraction of both resources is the WaCKi system (Bernardini et al., 2006). Specially challenging is, from both tasks, the acquisition of terminological lexicons for a given domain and this is the objective of the work described in this paper. Manual acquisition of terminology is of course feasible but costly and time consuming due to the need of highly skilled experts owning both a good knowledge of the domain and of the characteristics of terminology. An additional limitation of the manual acquisition is the extremely low level of agreement between the human extractors (as reported in Vivaldi and Rodríguez, 2007).

ISSN 1135-5948

In this paper we present an approach for extracting terminological information for a given domain using the Wikipedia (WP) as main knowledge source. The approach is domain and language independent and we have applied it to two languages (Spanish and English) and four domains (Medicine, Economy, Astronomy and Chemistry).

2 State of the art

Terms are usually defined as lexical units that designate concepts in a thematically restricted domain. A main problem concerning terms regards their detection; therefore, term extractors issue terms candidates (TC) instead of terms. There are some properties that a given TC must hold in order to be considered a term: a) unithood, b) termhood and c) specialised usage. The first characteristic refers to the internal coherence of a unit, the second to the degree a given candidate is related to a domain-specific concept and the latter to the specialised usage (general language versus specialised domain). It is clear that measuring such properties is not an easy task. They can only be measured indirectly by means of other properties easier to define and measure like frequency (of the TC itself, its components individually or in relation to general domain corpus), association measures, syntactic context exploration, highlighting and/or structural properties, position in an ontology, etc In some cases domain-specific features (as classic forms splitting in the medical domain) can be used but for the sake of domain independence of our approach we will not consider such features. Finding the appropriate measures, setting their weight assignments and a general way to combine them for a given task is still a research issue. Finally, it should be taken into consideration that, in a given language, terms are words; therefore they have the same formation rules.

In (Krauthammer et al., 2004), it has been mentioned that "terms identification has been recognized as the current bottleneck in text mining and therefore an important research topic in NLP". Such task is useful for a number of purposes: building terminological dictionaries; text indexing; automatic translation; improving automatic summarization systems, construction of expert systems, corpus analysis and, in general, whatever NLP application or task containing any domain specific component or needing domain specific tuning.

A term extractor can be viewed as performing a semantic annotation task because it intends to provide machine-usable information based on meaning. The way to attack the problem varies according the available resources for each language and domain. The pair English/Medicine constitutes a special case given that for this language/domain large resources (ontologies and term repositories) are available and can be used for reference. In this case, term extractor's process usually starts by chunking the text looking for the noun phrases and then trying to map each one to a reference list of terms. See (Krauthammer et al., 2004) or (Aronson et al., 2010) for a description of this type of tools and the necessary resources.

Those systems that cannot take profit of such repositories (most languages other than English, and domains other than Medicine), have to identify terms within text using other more complex procedures; involving linguistic/statistical strategies.

The results obtained using these techniques are not fully satisfactory as shown in (Cabré et al., 2001). Also, term extractors often favour recall over precision resulting in a large number of TC that have to be manually checked and cleaned. One of the reasons of such behaviour is the lack of semantic knowledge. Leaving aside Metamap, notable exceptions in the usage of this kind of information are TRUCKS (Maynard, 1999) and YATE (Vivaldi, 2001) that use as knowledge sources, UMLS and EuroWordNet respectively.

Using these kinds of lexico-semantic resources presents obvious limitations if we want to scale up to whatever language and domain. There are no resources of comparable size to UMLS for domains/languages other than Medicine/English and existing wordnets are general resources lacking of enough terminological coverage for most of the domains. The existence and coverage of other terminological resources as glossaries or lists of terms is very irregular. A promising alternative is the use of encyclopaedias as knowledge sources and the obvious choice is using the WP, as a free, high coverage in many domains, multilingual resource.

WP is by far the largest encyclopaedia in existence with more than 3.5 million articles in its English version contributed by thousands of volunteers. WP experiments an explosive

growing. There are versions of WP in more than 250 languages although the coverage (number of articles and average size of each article) is very irregular.

As a source of semantic information, WP can be exploited at least in two different ways: a) to obtain all the terms related to a given domain and b) given a "language for special purpose" corpus and a list of term candidates to identify those candidates belonging to the domain of such corpus. Both tasks can be seen as different ways to explore the WP, the former top down and the latter bottom up. Navigating WP top-down or bottom-up are not equivalent.

WP for a language is organized, as shown in Figure 1 into two connected graphs: the category graph (CG) and the page graph (PG). The whole article is assigned to one or more WP categories (through "Category links") in such a way that categories can be seen as classes that are linked to pages (belonging to the category). At the same time, a category is linked to one or more categories (super and sub categories) structuring themselves as classes that are also organized as a graph (see Zesch and Gurevych, 2007, for an interesting analysis of both graphs). This bi-graph structure of WP is far to be safe. Not always the category links denote belonging of the article to the category; the link can be used to many other purposes. The same problem occurs in the case of links between categories, not always these links denote hyperonymy/hyponymy and so the structure shown in the left of figure 1 is not a real taxonomy. Even worse is the case of inter-page links where the semantics of the link is absolutely unknown.

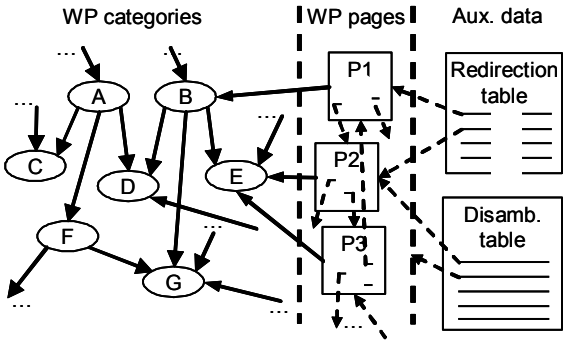

Figura 1: The graph structure of Wikipedia

The basic data unit of WP is the page or article, identified by a title and corresponding to a unique concept. Nodes of PG correspond to WP pages. There are, however, in WP several types of special pages: "Redirect pages", i.e. short pages which often provide equivalent names for an entity, and "Disambiguation pages", i.e. pages with little content that links to multiple similarly named articles. Other types of links (external links, interwiki links, etc.) are not considered in this work.

While edges between categories usually (but not always) have a clear semantics (hypernymy, hyponymy), edges between pages lack tags or explicit semantics. Also, some categories are added to WP by convenience for structuring the database or due to its encyclopaedic character (e.g. "scientists by country", "Chemistry timelines" or "Astronomical objects by year of discovery" among many others). Other categories are used temporally for monitoring the state of the page (e.g. "All articles lacking sources", "Articles to be split" ...), we name these categories "Neutral Categories". Due to this, it becomes difficult just navigating through its structure, to discover which entry belongs to which domain.

WP has been extensively used for extracting lexical and conceptual information: (Ponzetto and Strube, 2008) and (Suchanek, 2008), build or enrich ontologies from WP, (Milne et al., 2006) derive domain specific thesauri, (Atserias et al., 2008) produce a semantically annotated snapshot of EWP, (Medelyan et al., 2008; Mihalcea and Csomai, 2007 and Wu et al., 2007) perform semantic tagging or topic indexing with WP articles. Closer to our task are the works of (Toral et al., 2006) and (Kazama and Torisawa, 2007) which use WP, particularly the first sentence of each article, to create lists of named entities[1]. Other systems exploit the multilingual information of WP (Erdmann et al., 2008; Hecht and Gergle, 2010 are among them). See (Medelyan et al., 2009) or (Gabrilovich and Markovitch, 2009) for excellent surveys.

Extracting information from WP can be done easily using a Web crawler and a simple HTML parser. The regular and highly structured format of WP pages allows this simple procedure. There are, however, a lot of APIs providing easy access to WP online or to the database organized data obtained from WP dumps. Some interesting systems are Waikato's

[1] Although the overlap between name entities and terms is assumed to null, the techniques applied can be similar.

WikipediaMiner toolkit, U. Alicante's wiki db access, Strube and Ponzetto's set of tools, Hecht, Gergle's wikAPIdia, Iryna Gurevych' JWPL, etc. We have used this later resource for our research.

3 Methodology

Our system is intended to be language and domain independent and uses WP as a unique resource. Therefore, for any language to be considered the only limitation, regarding both qualitative and quantitative, comes from the WP for such language.

Two strategies can be followed in our proposal, depending on whether or not we dispose of a list of TCs. If we lack of such resource we proceed top down through several iterations from top categories of WP, we name this process TC production. If we dispose of a candidates list we proceed bottom up from its members. In this later approach the result will be the list enriched by attaching termhood score to all its members, we name this process TC validation. In the former approach also a ranked list of TC is provided this time starting from scratch. For evaluation these TC lists will be compared with the lists produced, using alternative knowledge sources, by others term extractors. In our work we have used YATE.

3.1 Obtaining terms from Wikipedia

The basis of our two approaches consists of locating two subgraphs, *CatSet* in CG, and *PageSet* in PG having a high probability of referring to concepts in the domain, our guess is that the titles of the members of both sets are terms. We score the categories in *CatSet* and the pages in *PageSet* using as knowledge sources the edges incident to the corresponding nodes in both graphs. The process is iterative and at each iteration scores of pages is leaked to the categories they belong to, and scores of categories to the pages they contain. In each iteration the less scored units (pages or categories) are removed.

3.1.1 Term candidates production

A preliminary version of the topdown approach, limited to the medical domain has been described and evaluated in Vivaldi and Rodríguez, 2010. We describe it very briefly. First we have to choose in the CG of WP the right tops for the domain. The process is automatic and may use different words to define a domain for looking in CG and PG. For instance "Medicine" directly corresponds to a category in CG while "Economy" corresponds to a page that is linked to two categories in CG. A different case is domain like "computer science" that may be defined using two words: "hardware" and "software". From this top set CG is traversed following the subcategory links, avoiding cycles, filtering out neutral categories and categories placed in the CG above the domain tops. The categories in this initial set are scored, using only the links to parent categories, as shown in (1), then all categories with scores less than 0.5 are removed from the set resulting in our initial set, $CatSet_0$.[2]

$$score_{cat} = \frac{|\text{parents}^{ok}_{cat}|}{|\text{parents}^{ok}_{cat}| + |\text{parents}^{ko}_{cat}|} \quad (1)$$

where

$$\begin{cases} \text{parents}^{ok}_{cat} = \text{set of parent categories under domain tops} \\ \text{parents}^{ko}_{cat} = \text{set of parent categories outside domain tops} \\ \text{parents}^{neutral}_{cat} = \text{set of parent categories neutral} \end{cases}$$

From each category $CatSet_0$ the set of belonging pages is collected in $PageSet_0$. Each category is scored according to the scores of the pages it contains and each page is scored according both to the set of categories it belongs to and to the sets of pages pointing to it and pointed from it. Three thresholding mechanisms are used: *Microstrict* (accept a category if the number of member pages with positive score is greater than the number of pages with negative score), *Microloose* (similarly with greater or equal test), and Macro (instead of counting the pages with positive or negative scoring we use the components of such scores, i.e. the scores of the categories of the pages).

$$score_{page} = comb(score^{cats}_{cpagt}, score^{input}_{pag}, score^{output}_{pag}) \quad (2)$$

where

$$\begin{cases} score^{cats}_{cpagt} = \frac{\sum_{\forall cat_k \in cats(page)} score_{cat}}{|cats(page)|} & \text{with cats(page) = set of categories of page} \\ score^{input}_{pag} = \frac{\sum_{\forall p_i \in input(page)} score_{p}}{|input(page)|} & \text{with input(page) = set of pages pointing to page} \\ score^{output}_{pag} = \frac{\sum_{\forall p_o \in output(page)} score_{p}}{|output(page)|} & \text{with output(page) = set of pages pointed from page} \end{cases}$$

and *comb* is a combinaton function of their arguments

[2] We use the subindex to refer to the iteration number.

$$score_{cat} = comb(score_{cat}^{strict}, score_{cat}^{loose}, score_{cat}^{micro}) \quad (3)$$

where

$$\begin{cases} score_{cat}^{strict} = \dfrac{count(score_{page} > 0.5)}{|\ pages(cat)\ |} & \text{with pages(cat) = set of pages of cat} \\ score_{cat}^{loose} = \dfrac{count(score_{page} \geq 0.5)}{|\ pages(cat)\ |} & \text{with pages(cat) = set of pages of cat} \\ score_{cat}^{micro} \dfrac{\sum score_{page}}{|\ pages(cat)\ |} & \text{with pages(cat) = set of pages of cat} \end{cases}$$

and *comb* is a combinaton function of their arguments

Then, we recursively explore each category and repeat the same process again. This way the set of well scored pages and the set of well scored categories reinforce each other. Less scored categories and pages are removed from the corresponding sets. As seen in (2) and (3), a combination function is used for computing the global score of each page and category from their constituent scores. Several voting schemata have been tested. We choose a DT classifier using the constituent scores as features. A pair of classifiers, $isTerm_{cat}$ and $isTerm_{page}$, independent of language and domain, were learned, as described in section 4. The process is iterated, leading in iteration i to $CatSet_i$, $PageSet_i$, until convergence[3].

3.1.2 Term candidates validation

For following the validation approach, a list of TCs is needed. For getting it we used the TC extraction module in YATE, the above mentioned hybrid term extractor. The reference vocabulary is also used to compare YATE performance using its own semantic analyzer module and the WP-based scorer presented here. The methodology is shown in Figure 2.

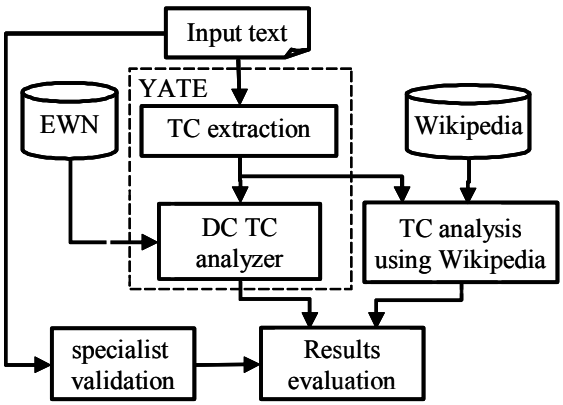

Figura 2: Evaluation of term extraction results

[3] In all the cases, convergence, i.e. the size of the sets is stable, was reached in less than 7 iterations.

Our aim is to attach to each TC a termhood[4] score using WP as unique knowledge source. Only TC occurring as category or page titles in WP are, thus, scored.[5] Our method consists of the following steps:

1) $CatSet_0$ is computed as in the first approach.
2) The set of top categories in WP is computed as well as in the first approach.
3) In the case TC corresponds to a category title in WP its DC is computed as in (1).
4) In the case a TC, t, corresponds to a page title in WP, we obtain its corresponding page, P_t, and its sets of categories, input and output page links and from these we obtain the DC as in (2).

Using the information collected during this exploration we identify at least three ways to calculate the DC for a given term t, using only the page (category link) and CG information and two more using the input/output links:

1. DC based on the number of paths.

$$DCnp(t) = \frac{NP_{domain}(t)}{NP_{total}(t)} \quad (4)$$

where $NP_{domain}(t)$ number of paths to the domain category tops
$NP_{total}(t)$ number of paths to whatever top

2. DC based on the number of single steps within the paths.

$$DClp(t) = \frac{NS_{total}(t) - NS_{domain}(t)}{NS_{total}(t)} \quad (5)$$

where $NS_{domain}(t)$ number of steps to the domain category tops
$NS_{total}(t)$ number of steps to whatever top

3. DC based on the average length paths.

$$DCalp(t) = \frac{ALP_{total}(t) - ALP_{domain}(t)}{ALP_{total}(t)} \quad (6)$$

where $ALP_{domain}(t)$ average path length to the domain category tops
$ALP_{total}(t)$ average path length to whatever top

Figure 3 shows a simplified sample of the WP organization around the Spanish term

[4] i.e. not only domain appropiateness but also term appropiateness are considered.

[5] YATE's Domain Coefficient, DC, measures the domainhood of a TC, just as $score_{cat}$ and $score_{page}$ do in our approach.

sangre (blood). The domain category chosen as DB is Medicine (shaded oval). The figure shows also how the above DCs are applied to this term.

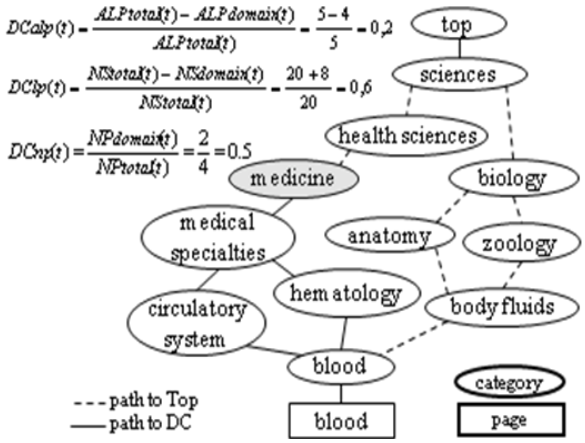

Figure 3: Usage of the DCs to the term blood.

The latter two factors, DC_{in} and DC_{out} are computed as in (2). For each candidate, *t*, existing as page in WP we obtain the corresponding page P_t (performing a disambiguation process when the page is, or points to, a disambiguation page).

For combining the results of these methods we have learned another decision tree classifier, $isTerm_{TC}$. We have used as features the 5 DC methods defined above, the syntactic class of *t* (noun, noun-adjective, etc.) and the type of P_t (category, standard page, disambiguation page).

4 Experiments and evaluation

As first step we have learned the three classifiers needed by our system, namely $isTerm_{cat}$, and $isTerm_{page}$, for the extraction task and $isTerm_{TC}$ for the validation one. We have used the Weka toolbox (Witten and Eibe, 2005). As the features we are mainly based on the structural properties of WP (with the only exception of the syntactic category) we decided to learn classifiers independent of language and domain[6], i.e. only three classifiers in order to reduce as much as possible the dependence on additional resources not always available for all the domain and languages. As learning vocabulary we used SNOMED-CT (http://www.ihtsdo.org/). We selected from it all the terms occurring as titles of categories or pages in Spanish WP. We reserved half of this material for additional testing and using the rest for learning as positive examples. As the coverage of SNOMED-CT is excellent we used all TC not occurring in this set as negative examples in the case of $Term_{page}$ and $isTerm_{TC}$. In the other case we used as negative examples the instances not occurring in SNOMED-CT and having no categories in $CatSet_0$. The resulting accuracy for the three classifiers is over 85%.

We performed first a set of experiments following the production approach. The results are presented in Table 2. For each language and domain the initial number of categories and the final size of $CatSet_i$ are presented. Both category names and page names over a threshold are considered terms. For each case the number of terms and the precision are included in the table. For the case ES/Medicine, precision has been computed using the reserved subset of SNOMED-CT (last column of Table 2). In the other cases evaluation has been performed comparing our results with the less reliable Domain Codes attached to WN (and EWN) synsets (see Magnini and Cavaglià, 2000).

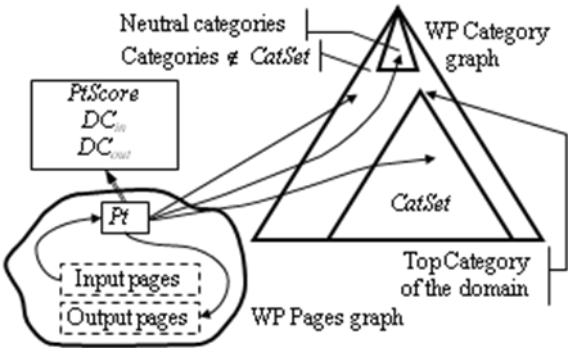

Figure 4: WP additional filtering

Another set of experiments was conducted following the validation approach. In this case the term extraction mechanism has been compared with an existing tool (YATE). Our evaluation method consists of comparing the performance of WP as a source of semantic knowledge with an equivalent knowledge obtained from EWN using YATE. In what follows we will use $YATE_{EWN}$ and $YATE_{WP}$ for referring to the original system and system resulting using WP.

In this case, we used two documents from the IULA's LSP Corpus (see Vivaldi, 2009); the first one from the medical domain and the second about Economics. Both documents

[6] With the only change of transforming the syntactic categories into bag of pos.

amount about 150 Kwords and has been segmented, tokenized and pos-tagged. We evaluate the results using the standard measures of precision and recall.

For evaluating our approach we performed two set of tests: from one side we evaluate the behaviour of the basic DCs —as defined in (4), (5) and (6)— and from the other side we evaluate the behaviour of the system using the additional information got from input/output links. Besides evaluating all patterns together but in the former case we evaluated also each pattern individually. The results for both domains are shown in Figure 5. For Medicine we show the results using the DC for each pattern (noun, noun-adjective and noun-prep-noun, all patterns together and all patterns using $isTerm_{TC}$) while for Economics only the results for the global performance is presented.

As can be seen examining both figures the results obtained using YATE is slightly better in Medicine but worst in Economics. Such behaviour is due to the different coverage of EWN in both domains. The results for each pattern may be summarized as follows:

- Noun: In Medicine, $YATE_{EWN}$ performs better than $YATE_{WP}$; the difference varies among 10 % (DC_{np}) and 20 % (DC_{apl}) for recall values lower than 30%. In Economics, $YATE_{WP}$ performs better for low recall values but worst for medium recall values. In spite of such difference it should be mentioned that DC_{np} ranks very well TC not existing in EWN, like *recesión* (recession) and, so, not detected by $YATE_{EWN}$.
- noun-adjective: in this case for Medicine the behaviour is similar to the nouns for DC_{np} and DC_{pl} but in Economics the performance is similar to nouns. Terms candidates like *historia clínica* (medical record), or *sector público* (public sector) are classified better than by using $YATE_{EWN}$.
- noun-prep-noun: in this case the performance of all $YATE_{WP}$ based CDs is better than those using $YATE_{EWN}$ for both domains. WP contains many terminological units like *protocolo de tratamiento* (treatment protocol) and *capacidad de producción* (productive capacity) obtain the maximum value with DC_{pl} but very low values using $YATE_{EWN}$ (the full strings are not in EWN and their components in isolation are not terminological).

As usual in this kind of evaluations the list of terms evaluated by the specialists is very troublesome due to completeness and criteria differences. The list of tagged terms by the specialists leaves aside some actual terms. See for example the cases of *epitelio* (epithelium) and *externalidad* (externality) that are well detected and ranked but they are not tagged as term by specialist; therefore, are considered as mistakes.

Domain			Chemistry		Astronomy		Medicine		
Language			EN	ES	EN	ES	EN	ES	ES*
Initial Categories			188374	2070	188816	44631	124503		2431
#Categories after pruning			1334	557	790	143	882		904
Stable iteration[7]	Categories		680/61	38/9	75/1	8/0(1)	46	159	11
	Precision		96.7	44.4	0.0	0.0	97.8	79,9	63.6
	Pages found	939	725/136	415/73	86/13	5350/704	856/156	6189	6189
		738	454/87	278/46	53/6	3761/462	522/85	5554	5554
	Prec. [%]	61.3	50.7	47.9	61.6	72.6	85.2	63.0	63.0
		61.5	52.9	50.0	50.0	72.1	87.1	67.0	67.0

Table 2. Results of the experiments (* evaluated using SNOMED-CT)

[7] Both "Pages found" and Categories rows include two values: *X*/*Y* where *X* is the total number of CATs found in WP and searched in WN and *Y* the total numbers of CATs found in the Domains code attached to WN. The precision values have been calculated using *Y*.

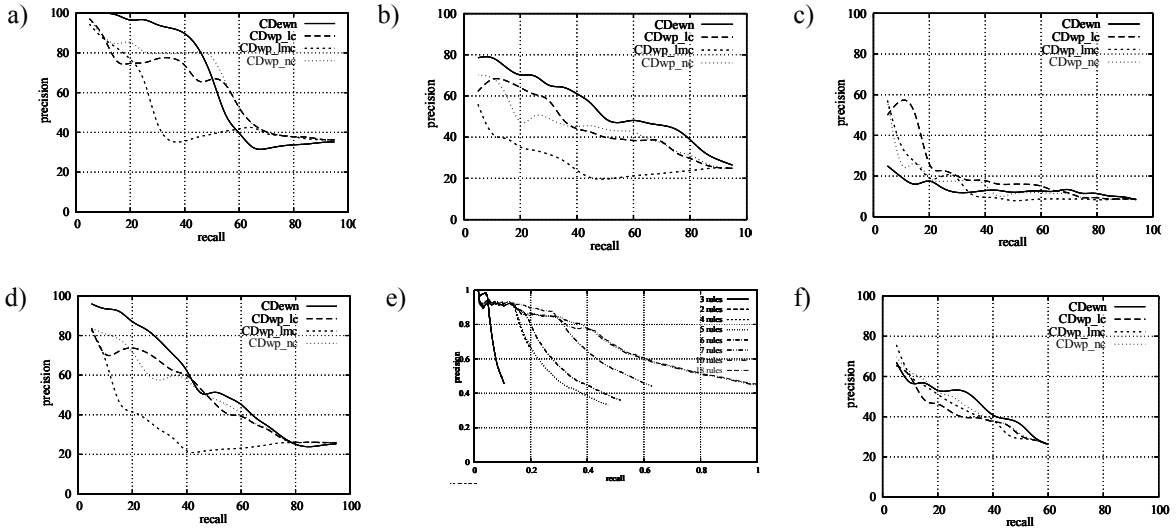

Figure 5: Results of term extraction for medicine (a: noun pattern, b: noun-adjective pattern, c: noun-prep-noun pattern, d: all patterns and e: all patterns using decision trees) and economics (f: all patterns).

5 Conclusions and future work

In this paper we present a system for obtaining the terminology of a domain using as unique resource the category and page structures of WP in a language independent way. The system has been tuned, applied and evaluated in two different scenarios: production and validation of a set of term candidates. They have been applied to several domains and languages showing good performance compared with the state-of-the-art in TE.

In the next future we plan: i) to apply our method to other languages and domains, specifically to apply our system to the whole set of Magnini's domain codes; ii) to use not only the category and page graphs but also the text contained in the best scored pages for improving the recall of term selection, especially in the production procedure.

6 Bibliography

Aronson A. and F. Lang, 2010. An overview of MetaMap: historical perspective and recent advances. JAMIA 2010 17:229-236.

Atserias J. H. Zaragoza, M. Ciaramita and G. Attardi, 2008. Semantically Annotated Snapshot of the English Wikipedia. Proceedings of the 6th LREC Conference.

Barrón-Cedeño A., Sierra G., Drouin P., Ananiadou S. 2009. An improved automatic term recognition method for Spanish. In Proceedings of the 10th CICLing Conference, pages 125-136, Mexico.

Bernardini, S., M. Baroni y S. Evert. 2006. A WaCky Introduction. Wacky! Working papers on the Web as Corpus, pages 9-40, Bologna: Gedit.

Erdmann M., Nakayama K., Hara T. and S. Nishio, 2008. Extraction of Bilingual Terminology from a Multilingual Web-based Encyclopedia. Journal of Information Processing. No. 3: 564-575.

Gabrilovich E. and S. Markovitch, 2009. Wikipedia-based Semantic Interpretation for Natural Language Processing. Journal of Artificial Intelligence Research 34:443-498.

Hecht B. and D. Gergle, 2010. The Tower of Babel Meets Web 2.0: User-Generated Content and its Applications in a Multilingual Context. In Proceedings of the 28th ACM Conference on Human Factors in Computing Systems: 291-300, Atlanta, GA.

Kazama, J. and K. Torisawa, 2007. Exploiting Wikipedia as External Knowledge for Named Entity Recognition. Proceedings of the EMNLP-CoNLL Conference: 698-707.

Krauthammer M. and G. Nenadic, 2004. Term identification in the Biomedical Literature. Journal of Biomedical Informatics. Vol. 37(6):512-526.

Maynard D., 1999. Term recognition using combined knowledge sources. PhD Thesis. Manchester Metropolitan University.

Magnini B. and G. Cavaglià, 2000. Integrating Subject Field Codes In WordNet. In Proceedings of the 2nd LREC International Conference: 1413-1418, Greece.

Medelyan O., David N. Milne, C. Legg and I. H. Witten (2009). Mining meaning from Wikipedia. International Journal of Human-Computer Studies. 67(9): 716-754.

Medelyan O. I. H.. Witten, and D. Milne, 2008. Topic indexing with Wikipedia. In Proceedings of Wikipedia and AI workshop at the AAAI-08 Conference. Chicago, US.

Mihalcea R. and R. Csomai, 2007. Wikify!: linking documents to encyclopedic knowledge. Proceedings of CIKM 233-242.

Milne D. D. Milne, D. Medelyan and I. H. Witten, 2006. Mining Domain-Specific Thesauri from Wikipedia: A case study. IEEE/WIC/ACM International Conference on Web Intelligence, WI'06: 442-448, Hong Kong.

Pazienza M.T., Pennacchiotti M. and F. M. Zanzotto, 2005. Terminology Extraction: An Analysis of Linguistic and Statistical Approaches. Studies in Fuzziness and Soft Computing 185, pages 255-279.

Ponzetto P. and M. Strube, 2008. WikiTaxonomy: A large scale knowledge re source. In: Proceedings of the 18th European Conf. on Artificial Intelligence, Greece: 751-752.

Suchanek F., 2008. Automated Construction and Growth of a Large Ontology. PhD-Thesis. Saarbrücken University, Germany.

Toral, A. and R. Muñoz, 2006 A proposal to automatically build and maintain gazetteers for Named Entity Recognition using Wikipedia. In Proceedings of the Workshop on New Text, 11th Conference of the EACL. Pages 56-61. Trento, Italy.

Vivaldi J. and H. Rodríguez, 2010. “Finding Domain Terms using Wikipedia”. In Proceedings of the 7th LREC International Conference: 386-393, Malta.

Vivaldi J. and H. Rodríguez, 2008. Evaluation of terms and term extraction systems. A practical approach. Terminology 13(2): 225-248. John Benjamins.

Vivaldi Palatresi, Jorge (2009). Corpus and exploitation tool: IULACT and bwanaNet. In Proceedings of *CILC-09*: 224-239, Spain.

Zesch T. and I. Gurevych, 2007. Analysis of the Wikipedia Category Graph for NLP Applications. In Proceedings of the TextGraphs-2 Workshop: pages 1-8.