Supervised Learning Algorithms Applied to Terminology Extraction

Rogelio Nazar and Maria Teresa Cabré

Institut Universitari de Lingüística Aplicada
Universitat Pompeu Fabra
Roc Boronat 138, 08018, Barcelona.
{rogelio.nazar,teresa.cabre}@upf.edu

Abstract. In this paper we present a new terminology extraction system based on supervised statistical learning algorithms, which are characterized by having a training phase with a controlled exposure to both positive and negative examples prior to the actual categorization. Contrary to the vast majority of the term extractors reported in the literature, our proposal is based on implicit knowledge rather than handcrafted explicit rules. Given a list of terms from some domain and language plus a general language reference corpus, we developed a methodology for terminology extraction and implemented it as a web application that is already available online. This tool is flexible enough to operate in different languages and domains and, as a sort of lifelong learning algorithm, it turns terminology extraction into a collaborative effort, where all users benefit from the training conducted by each individual.

Keywords: computational terminography, machine learning, quantitative linguistics, terminology extraction.

1 Introduction

The development of computational algorithms for the automatic extraction of terminology from LSP corpora has been established as an autonomous field of research for the past two decades, and yet we have not witnessed any major change in the basic conceptions and methodology. The vast majority of authors have implicitly or explicitly stated that terminology extraction is a language- and domain-specific enterprise [3, 5, 14]. Statistical algorithms have been applied to the task, but the general opinion expressed by authors (see Section 2) is that statistical algorithms are not precise enough and could benefit from explicitly coded linguistic rules. Wrong assumptions have had a negative impact on the research, for instance, the idea that statistically based term extractors can only work on massive amounts of data and are not useful to extract low frequency terms. While it is true that there are a number of statistical term extraction systems that have these limitations, this does not apply to statistical methods in general.

In this paper we would like to introduce a different view, from the angle of statistical learning algorithms. We hope that our contribution will stimulate

Aguado de Cea et al. (Eds.): Proceedings of the 10th Terminology and Knowledge Engineering Conference (TKE 2012), pp.209-217. 19-22 June 2012, Madrid, Spain

debate and suggest a new direction in the discussion on the design of statistically based term extraction systems, where we would have the possibility of working with samples of any size and be able to extract low frequency terms (hapax legomena and dis legomena), thus considerably expanding recall.

In this line, we propose a new algorithm whose main advantage, in comparison with others reported in the literature, is its flexibility to adapt to new languages and domains because there is no need to introduce explicit linguistic or ontological knowledge into the system. Clearly, the algorithm needs some kind of linguistic knowledge to produce results, since it uses POS-tagged text and syntactic chunking, but in the form of implicit knowledge, which is a fundamental difference. The system is based on learning by examples, and this means that there is no need for experts to code knowledge in the form of explicit rules, which is very costly. POS-tagging, for instance, in an implementation such as Schmid's [22], is based on a training phase where the program learns to generalize from a sample of hand-tagged text. Of course, producing a sample of POS-tagged text that is large enough to train the tagger is still an important human effort if it must be done by hand, yet it is in a completely different proportion to the effort needed to hand-craft large coverage disambiguation rules. Similarly, the point of departure of our term extractor is a learning phase where the algorithm is trained with examples of both terminological units as well as general language text. With this material, the algorithm develops a statistical model with an abstraction of the main characteristics of both samples. This model is then used in a test phase where the algorithm processes new LSP corpora in the same domain field and language as in the training and extracts and ranks term candidates. In addition, we have envisaged the implementation of this algorithm as a tool for terminology as a collective, and therefore network-based, effort. The algorithm is now implemented and running on a web server[1] and can be accessed for demo purposes. What is interesting about this implementation is that as more users train the program, it progressively extends its capabilities and this cumulative "experience" is available to the rest of the users.

The paper is organized as follows. After a brief comment on related work, we describe the algorithm in more detail, followed by an evaluation and subsequent discussion of the results obtained in an experiment involving the extraction of terms from a corpus consisting of papers from the *Computational Linguistics* journal[2].

2 Related Work

The origin of most of the current work related to terminology extraction can be traced back to more general proposals in the field of computer-assisted terminol-

[1] This extraction algorithm has been recently added to the online terminology management tool Terminus (http://terminus.upf.edu) and can be freely accessed for demo purposes. Users can train the program to extract terms in different languages and domains or make use of the training previously undertaken by other users.

[2] http://www.mitpressjournals.org/loi/coli [Accessed on: 22/01/2012]

ogy processing [20], which have evolved in parallel with the advances in related fields such as information retrieval [23, 21], corpus linguistics [24] and theoretical research in terminology [4]. The technological developments in computer science in the late eighties and the widespread use of computers in universities and linguistics departments led to a proliferation of proposals in this field and this growing interest has not yet shown signs of decay.

Space limitations prevent us from offering even a brief description of some of the best-known contributions (see [16] for an overview). In broad terms, we could separate a first branch of authors with a tendency to include different degrees of linguistic knowledge, which can include lexical, morphological, syntactic and semantic levels [1, 15, 2, 13, 3, 5, 14, 25]. With respect to the more quantitatively-oriented philosophy, most authors have opted to focus on the associational strength of the different components of multi-word terminology [6, 7, 9–11, 17] but there are different approaches where the distribution of terms in different collections of documents is what is taken as a clue for the extraction [12, 8]. The approach to terminology extraction based on supervised learning algorithms has been attempted before [18], training a classifier with a corpus that contains the terms hand-tagged by experts. This method, however, requires large amounts of data and suffers with low frequency terms.

We have not been able to find any study with an exhaustive and up-to-date comparison of the performance achieved by each different method, but this is not surprising given the difficulty of such an enterprise, since there are different aspects that should be compared (efficiency, accuracy, economy, etc.) and each term extractor has a variety of execution parameters that must be taken into account. More crucially, we find that there is a lack of studies on measures of inter-annotator agreement in the task of evaluating or training terminology extraction systems. Calculating the intersection between lists of terms manually extracted by terminologists or domain experts from LSP corpora would be, undoubtedly, a very interesting line of future work. If it turns out that there is too little consensus between humans, then this would go against the most basic assumptions behind the whole terminology extraction effort.

3 Methodology

As already mentioned in the introduction, the training of the algorithm is conducted basically with two types of materials: examples of terms from the domain of interest and a reference corpus of general language, which represent positive and negative examples of terms. The learning is conducted at three levels: syntactic, lexical and morphological. The list of positive examples is processed with Schmid's [22] POS-tagger. This is to calculate the frequency distribution of the POS-tag sequences and use it to develop a syntactic model of the terms. The lexical model simply accounts for the frequency of the lexical units within the terms (both forms and lemmata) and, finally, the morphological learning is conducted by extracting, from each word type, initial and final character n-grams ($1 \leq n \leq 5$).

In the test phase, the algorithm accepts new text as input and produces a first list of candidates by extracting those POS-tag sequences that were frequent in the training (which are normally noun phrases). The final terminological score of a term candidate is calculated in the same way for all the levels of the training. The basic idea is to weight better those elements which have a significant frequency in the LSP training material with respect to the general language corpus. Patry & Langlais [18] offer a similar approach for learning the patterns, but instead of relaying on the frequency of the patterns in the training set, they resort to a language model arguing that, in this way, they can generate patterns that were not present in the training set. A further exploration of this alternative, however, is left future work. As shown in Equation 1, for each term candidate c found in an analyzed corpus, we calculate the frequency of the features from each level i of the learning. The symbol $f_o(c_i)$ represents the observed relative frequency of feature i in the training corpus, while $f_e(c_i)$ is the relative frequency of the same element in the reference corpus and $f_a(c)$ the actual frequency of c in the analyzed text.

$$T(c) = (\prod_{i=1}^{|c|} \frac{f_o(c_i)}{f_e(c_i) + 1}) f_a(c) \tag{1}$$

The collaborative approach is a very important aspect of this term extractor and probably one of its most interesting features. The fact that the algorithm is implemented as a web application allows a community of terminologists to share knowledge acquired by the program in each training phase. Different users are currently training this program in different languages and domains. As a consequence, our program is constantly improving in both precision and recall, as a sort of lifelong learning algorithm. However interesting this might be, it is an aspect that has not been considered for the evaluation presented in the next section because of the need to separate the performance of the algorithm from the contribution made by its users.

4 Results and Evaluation

In this section we report the results obtained with this algorithm in an experiment involving the extraction of terms from the domain of computational linguistics, where the authors feel confident enough to act as specialists to judge the acceptability of the proposed term candidates. As a corpus, we used the papers that appeared in the *Computational Linguistics* journal in the years 2010 and 2011.

As the experimental set-up demands a training set and a test set, we separated part of the corpus (papers published in the year 2010) as the learning set and the rest (those published in 2011) as the test set. The approximate extension of the training set corpus is 300,000 words and the test set corpus contains around 340,000. As a reference corpus, we used a collection of English press articles with an extension of two million words downloaded from the Leipzig Corpora Collection [19].

To train the algorithm, we extracted term candidates from the training set corpus using the same algorithm (trained for a different domain), as well as other methods such n-gram frequency lists and word-association measures (described in more detail later when we introduce our baseline algorithms). From the set of all term candidates obtained with these methods, we manually validated 800 terms using our own expert knowledge in the field, and then trained the algorithm with this list. This list of training terms that we selected from the training corpus comprises both single and multiword expressions. A few examples of them are shown in Table 1. Table 2, in turn, shows a fragment of the output of the training phase, which consists in the selection of the most frequent syntactic patterns among the set of training terms or, strictly speaking, the most frequent sequences of POS-tags, with N for noun, J for adjective and V for verb. As was to be expected, noun-noun and adjective-noun are among the most frequent sequences.

Table 1: Examples of terms from the domain of computational linguistics used for the training phase

Examples of training terms
bilingual word alignment
corpora
discriminative learning
iterations
linear models
markov models
maximal nodes
memory store
node
nouns
rule
selectional preferences
sentence length
source words
surface realization
target phrase
test data
training text
word sense disambiguation

With the training process finished, we subsequently processed the corpus from 2011 and obtained a ranked list of candidates from which we only evaluated the first 1500 positions. Remarkably, among these there were only 93 terms in common with the training set list.

The evaluation is conducted in comparison with three baseline algorithms inspired on classical approaches to the problem. One is the extraction of bigrams

Table 2: Syntactic patterns of the terms used for training, sorted by decreasing order of frequency

Syntactic patterns of the training set	
N N	303
J N	215
N	164
J N N	26
V N	25
N N N	22
J J N	10
J	6
V	5
N V	4
J N N N	3

by frequency (the 1500 most frequent bigrams) filtering them with a stoplist of frequent and uninformative words. The other two are lists of bigrams, this time obtained not by frequency but by mutual information (MI) and chi-squared statistics, with a frequecy threshold set to 5 to compensate for the fact that these measures tend to promote low frequency bigrams. Evaluations of term extraction systems are commonly reported in terms of precision and recall or with cumulative precision plots. In the present evaluation we opted for the second, as we had no reliable estimation of the total number of terminological units present in the test corpus. As shown in Figure 1, the results obtained with our method largely outperformed those obtained by the three baseline algorithms, achieving 85% precision in the first 200 positions and 75% in the first 400. Table 3 shows some of the candidates selected by the algorithm and a binary value indicating that the candidate is accepted or rejected. The first 15 positions are shown in panel A and positions 200 to 215 in panel B. Naturally, the main argument to accept or reject a term candidate is to decide if it would be a useful entry in a computational linguistics glossary.

The bigrams sorted by frequency are half-way between those obtained by our algorithm and those obtained by MI and chi-squared, which are penalized by promoting bigrams which show a strong association but are not, however, terms, such as linguistic expressions (*bona fide, vice versa, submission received*), proper nouns (*Della Pietra, Eric Clapton*), wrongly segmented larger units (*vector machines* or *linear context-free* for *support vector machines* and *linear context-free grammar* or *linear context-free rewriting systems*, respectively), among others.

4.1 Conclusions and Future Work

In this paper we have presented a new method for terminology extraction based on learning by examples which is characterized by a considerable flexibility to adapt to new languages and domains. In its current design, the adaptability to

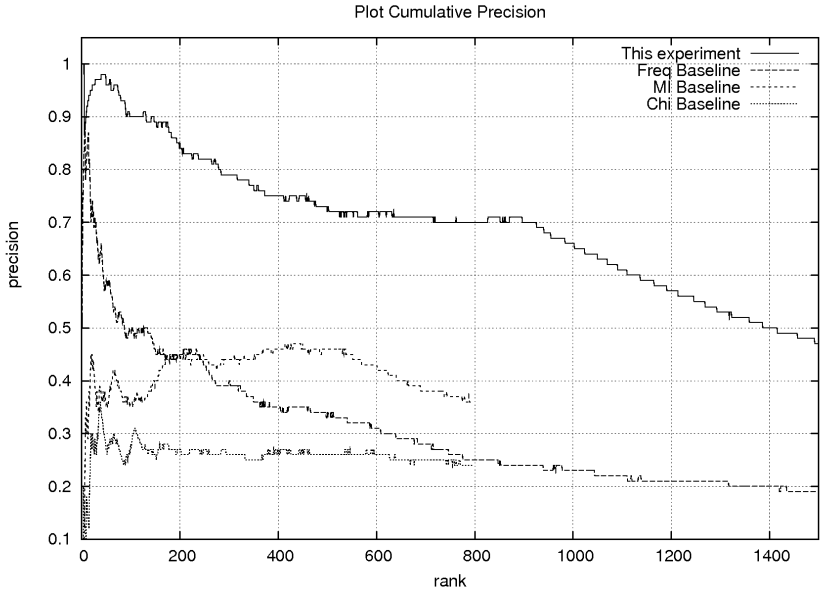

Fig. 1: Cumulative precision plot of algorithm and three baselines for the first 1500 term candidates.

Table 3: Examples of term candidates in different positions of the ranking

(a) First 15 positions

Rank	Candidates	Status
1	algorithm	1
2	corpus	1
3	model	1
4	annotation	1
5	node	1
6	language	1
7	method	0
8	word	1
9	grammar	1
10	parser	1
11	co training	1
12	linguistics	1
13	phrase	1
14	annotator	1
15	computational linguistics	1

(b) Positions 200 to 215

Rank	Candidates	Status
200	lcfrs grammar	0
201	lexical acquisition	1
202	s algorithm	0
203	corpus analysis	1
204	semantic class	1
205	long distance	0
206	npb node	0
207	language resource	1
208	corpus annotation	1
209	linguistics page	0
210	new node	0
211	parse algorithm	1
212	syntactic analysis	1
213	lexical dependency	1
214	annotation guideline	1
215	grammar development	0

different languages is conditioned by the difficulty involved in training a POS-tagger in the corresponding language. More empirical research is necessary in different languages before claiming the idea is language independent. However, it is interesting that in the languages where it has been tested so far, this knowledge has not been needed. Similar results have been achieved in experiments in different domains in Catalan, French, German and Spanish, and we plan to undertake experiments in more languages as more users from different parts of the world show interest in this project.

We see practical advantages in its application in the context of a tool for computer-assisted terminology processing, which entails diverse functions such as corpus constitution and analysis, and term database management. Our first user tests have shown that the training as well as the extraction can be conducted by users with a low level of computing skills. Furthermore, the algorithm is simple enough to make its implementation a rather straightforward programing exercise, and it is also computationally inexpensive, resulting in the fast execution needed in a web environment (both the training and the extraction of the experiment reported in this paper were completed in less than 10 seconds, with most of this time devoted to the POS-tagging of the text).

For future work, we plan to continue evaluating our method in different languages and domains, as well as trying to progressively include more learning features, such as user feedback and generalizations on lexical elements that tend to be present in both accepted as well as rejected term candidates.

Acknowledgments. This work has been made possible thanks to funding from project APLE (Spanish Ministry of Science and Education: Ref. FFI2009-12188-C05-01, subprogram FILO) entitled "Updating processes of the Spanish lexicon from the press" Period: 2010-2012. Project Leader: Dr. M. Teresa Cabré. We would like to express our gratitude to the Association for Computational Linguistics for their kind permission to use the Computational Linguistics Journal as a corpus. We also thank the anonymous reviewers for their insightful comments and Mark Andrews for his proofreading.

References

1. Ananiadou, S.: A Methodology for Automatic Term Recognition. Proceedings of Coling 1994, 15th International Conference on Computational Linguistics, Kyoto, Japan: 1034–1038 (1994)
2. Bourigault, D.: LEXTER, a Natural Language Tool for Terminology Extraction. Proceedings of the 7th EURALEX International Congress, Göteborg: 771–79 (1996)
3. Bourigault, D., Jacquemin C.: Term Extraction+Term Clustering: an integrated platform for computer-aided terminology. Proceedings of EACL 1999, Bergen: 15–22 (1999).
4. Cabré, M. T.: La terminología: representación y comunicación. Barcelona: Institut Universitari de Lingüística Aplicada (1999)
5. Collier, N., Nobata C., Tsujii J.: Automatic acquisition and classification of terminology using a tagged corpus in the molecular biology domain. Terminology, 7(2): 239–257 (2002)

6. Dagan, I., Church, K.: Termight: Identifying and Translating Technical Terminology. ANLC '94 Proceedings of the Fourth Conference on Applied Natural Language Processing, Association for Computational Linguistics: 3440 (1994)
7. Daille, B.: Approche mixte pour l'extraction automatique de terminologie : statistiques lexicales et filtres linguistiques. Doctoral Dissertation. Université Paris 7 (1994)
8. Drouin, P.: Term Extraction using non-Technical Corpora as a Point of Leverage. Terminology, 9(1): 99–117 (2003)
9. Enguehard, C., Pantera, L.: Automatic Natural Acquisition of a Terminology. Journal of Quantitative Linguistics 2(1): 27–32 (1994)
10. Frantzi, K.T.: Incorporating context information for extraction of terms. Proceedings of the Association for Computational Linguistics (ACL/EACL), Madrid: 501–503 (1997)
11. Frantzi, K., Ananiadou, S., Mima, H.: Automatic recognition of multi-word terms. International Journal of Digital Libraries, 3(2): 117–132 (2000)
12. Hisamitsu, T., Niwa, Y, Nishioka, S., Sakurai, H., Imaichi, O., Iwayama, M., Takano, A.: Term Extraction Using A New Measure of Term Representativeness. Proceedings of the Second International Conference on Language Recources and Evaluation (LREC 2000). Workshop Proceedings on: Terminology Resources and Computation, 13–20. May 29, Athens, Greece (2000)
13. Jacquemin, C.: Spotting and Discovering Terms through Natural Language Processing. MIT Press (2001)
14. Joan, A., Vivaldi, J., Lorente, M.: Turning a Term Extractor into a new Domain: first Experiences. Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008) Marrakech: 748–753 (2008)
15. Justeson J., Katz, S.: . Technical Terminology: some Linguistic Properties and an Algorithm for Identification in Text. Natural Language Engineering 1(1): 9–27 (1995)
16. Kageura, K., Umino, B.: Methods of Automatic Term Recognition. Terminology 3(2): 259–289 (1996)
17. Pantel, P., Lin, D.: A Statistical Corpus-Based Term Extractor. Proceedings of the 14th Biennial. Conference of the Canadian Society on Computational Studies of Intelligence, London, UK: Springer-Verlag: 36–46 (2001)
18. Patry, A., Langlais, P.: Corpus-Based Terminology Extraction. Proceedings of the 7th International Conference on Terminology and Knowledge Engineering, Copenhagen: 313–321 (2005)
19. Quasthoff, U., Richter, M., Biemann, C.: Corpus Portal for Search in Monolingual Corpora, Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006) Genoa: 1799–1802 (2006)
20. Sager, J.: A Practical Course in Terminology Processing. Amsterdam/Philadelphia: John Benjamins (1990)
21. Salton, G., McGill, M. J.: Introduction to Modern Information Retrieval. New Yor: McGraw-Hill (1983)
22. Schmid, H.: Probabilistic Part-of-Speech Tagging Using Decision Trees. Proceedings of the International Conference on New Methods in Language Processing, Manchester: 44–49 (1994)
23. Sparck Jones, K.: A Statistical Interpretation of Term Specificity and its Application in Retrieval. Journal of Documentation 28(1): 11–21 (1972)
24. Sinclair, J.: Corpus, Concordance, Collocation. Oxford University Press (1991)
25. Vivaldi, J.; Rodríguez, H.: Extracting Terminology from Wikipedia. Procesamiento del lenguaje natural 47: 65–73 (2011)