Using Morphological, Syntactical and Statistical Information for Automatic Term Acquisition [*]

Joana Lúcio Paulo (L^2F - INESC-ID / IST)
joana.paulo@inesc-id.pt
Margarita Correia (FLUL / ILTEC / SILEX)
margarita-C@netcabo.pt
Nuno J. Mamede (L^2F - INESC-ID / IST)
nuno.mamede@inesc-id.pt
Caroline Hagège (Xerox Research Centre Europe)
caroline.hagege@xrce.xerox.com

Rua Alves Redol 9, 1000-029 Lisboa, Portugal

Abstract. Terminologies are useful in all areas that use specialized languages. The development of terminologies is a hard work, when manually done. It can be assisted with tools to ease and improve the achievement of such a work. In this article, we present ATA, an automatic terms extractor using both linguistic and statistical information.

1 Introduction

In the last few years, computational linguists, applied linguists, translators, interpreters, scientific journalists and computer engineers, have been interested in automatically extracting terminology from texts. Different goals have led these professional groups to design software tools to directly extract terminology from texts, basically, all kind of Natural Language Processing (NLP), applications that work with specialized domains and that consequently need special vocabulary.

ATA, is an Automatic Term Acquisition System that processes technical texts and produces a list of noun phrases likely to be terminological units.

This article opens with some background knowledge. Then structural and functional aspects of ATA are presented: the architecture, the input and output data and the main process, responsible for the term's extraction. After this, we describe the evaluation process enumerating the expected results from the use of ATA. Finally we have a brief presentation of the future work and some notes.

2 Background

In this context, a *term* is a linguistic representation of a concept by means of a simple noun or a noun phrase [12]. We consider two term types: simple

[*] This paper has been partially supported by the Fundação Para a Ciência e Tecnologia under project number PLUS/1999/LIN/15150

and compound. Other phraseological structures characterizing some knowledge domains are not in the scope of ATA.

Simple terms consists of a single lexical unit, a graphical word. The complexity associated with the detection of this kind of terms arises from their unremarkable appearance. This means that there is no way for one to be distinguished from another, unless the system has a morphological structure analyser which can sort term-candidates by the occurrence of specific affixes or roots which is not the case of ATA.

Compound terms consists of more than one lexical unit (graphical form). Thus, they are less prone to ambiguity than simple terms. Nevertheless, they require a previous syntactical study to verify whether a set of words actually defines a term's syntactical structure.

According to several works [16, 13, 3, 12] all lexical units have an associated frequency corresponding to the number of times they appear in a corpus. Using this information we can decide whether a word can eventually be a term: items that are nouns and that appear more than a given number of times can be considered as candidates to be simple terms; words with other categories must be kept in order to complete the processing of compound terms.

Most systems designed for this kind of task take a plain text and extract from it a list of candidate terms. To make the terminologist's task easier, this list is provided with its context and assorted additional information (such as relative frequency for that word and for its root).

The most used techniques for this task are:

- **Statistical based systems:** detect lexical units whose frequency is higher than a given corpus-based threshold definition. The problem with this approach is that it fails to detect low-frequency terms.
- **Systems that use linguistic knowledge:** detect recurrent patterns from complex terminological units such as noun-adjective and noun-preposition-noun. Patterns to be detected are assumed to have been designed by linguists.
- **Hybrid systems:** start detecting some basic linguistic structures, such as noun or prepositional phrases, and then, after the candidate terms have been identified, the relevant statistical information is used to decide whether they correspond to a term. This will be our methodology.

The development of noun phrases extractors is a very delicate task constrained by robustness and accuracy.

Robustness is subject to a strong restriction: it can be used over a wide range of unrestricted texts gathered in large corpora. This means that it has to be domain-independent, that is, it cannot use any a priori semantic or conceptual information. From the point of view of the surface syntactic analysis, the extraction is more difficult, since the system is domain-independent because each domain can have specific restricted surface structures ([8] restricted the extraction to medical terms which have few possible nominal structures).

Accuracy is also an issue because the noun phrases extracted by the system are the candidate terms that will be proposed to the user building a domain's terminology.

The two most frequently used metrics in the evaluation of this type of system are recall and precision [14]. Recall is defined as the relationship between the sum of retrieved terms and the sum of existing terms in the document that is being explored. Precision accounts for the relationship between those extracted terms that are actually terms and the aggregate of candidate terms that are found. These metrics can be seen as the capacity to extract all terms from a document (recall) and the capacity to discriminate between those units detected by the system which are terms and those which are not (precision).

Systems based on linguistic knowledge tend to use noise and silence as a measure of efficiency. Noise attempts to assess the rate between discarded candidates and accepted ones; silence attempts to assess those terms contained in an analysed text that are not detected by the system. Errors in the assignment of morphological categories or syntactic analysis are also shared by these systems.

3 ATA's Structural and Functional Aspects

In this section we describe ATA's structure and functionality. First we take into account the application's structure, that is, the architecture of the system. Then the input and the output data are described. We kept the description of the main process for the next section.

3.1 ATA's Architecture

The used architecture (fig. 1) is similar to the proposed for simple terms in [7]. Taking the system as a whole, it first lemmatizes the text using an external dictionary. The resulting text is then passed to a post-morphological processor that detects and forms special groups according to recomposition and correspondence rules. The system groups the words in phrases (the phrase separators have been previously described in another external file). Before the main process begins, the text is sent to a syntactic analyser that, using a surface grammar groups the phrase constituents. Then the main process extracts the words candidate to terms. After this, a statistical-based process evaluates the candidate lists. The output is finally produced and sent to the system's user.

Lemmatization enriches each word with its morphological characterization. This step needs the dictionary file, which contains not only simple units (corresponding to graphical forms), but also complex units, such as prepositional and adverbial locutions. For this we use SMorph [1] that allows the construction of large dictionaries required for the linguistic analysis of texts. The user declares the dictionary by specifying five types of rules, which are converted into a compact binary file containing a finite state automata. The dictionary is used for generating all inflected forms of a lemma and for segmenting and analyzing a text.

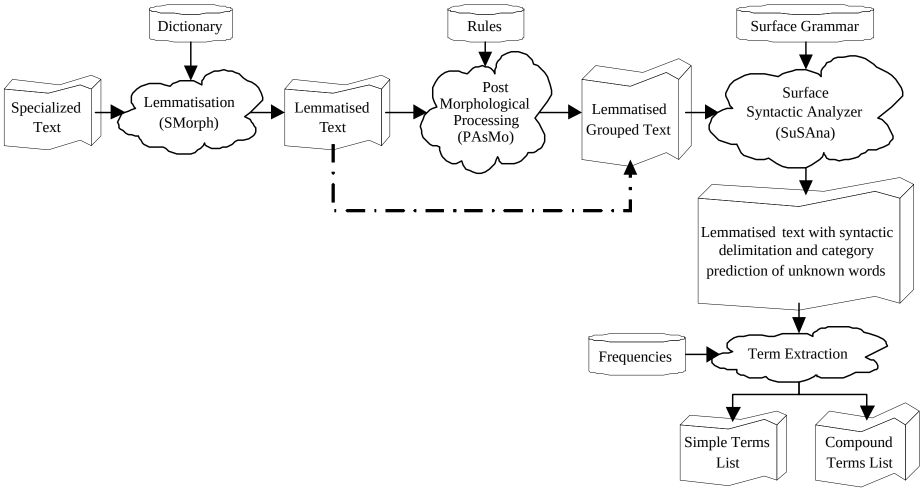

Fig. 1. Main Architecture.

Post-Lemmatization Process rewrites the text according to rules based on the morphological features of the words and breaking it into sentences (according to the chosen punctuation). This is done using recomposition and correspondence rules and a list of symbols to be used as breaks in the text. For this we use PAsMo [17]. that can rewrite dates, compound nouns, consecutive unknown words, numbers, and so on. PAsMo may also be used to translate tags, thus facilitating the interface with the syntactic analyzer. At this point we had to rebuilt an old system [9] from scratch in order to improve its efficiency and expressivity this is detailed on 4.

Syntactic Analysis is the stage where we analyse the text grouping the words of each phrase into syntactic chunks. This process is done using surface grammars. For this we use SuSAna [10, 2].

Term Extraction is the main process of the system that will be described latter on 5.

3.2 Input and Output Data

ATA's input data consists of plain text previously analysed both morphologically (in terms of words' categories) and syntactically. If we consider the larger system containing ATA, the input consists of a plain text, a dictionary, a set of rules defining word grouping (recomposition and correspondence rules), a surface grammar, and word's frequencies measured over general corpora.

Recomposition Rules are applied to the morphological analyzer's output and are used to change the segmentation in this input file. The rules are in the form

$A \rightarrow C$, which means that if we find a word (or a sequence of words) with the features declared on the rule's left hand side, A, then we can replace it with the rule's right hand side, C. For instance, a useful rule is one that takes a month, a number, "th" and another number and transforms this sequence into a date, i.e., it takes "December 6th 1978" and tags it is a date. This kind of rule may have variables and special operators that allow recursion on the rule's left side.

Correspondence Rules change the morphological descriptions associated with each word. The idea is the same, if we have something that matches the rule's left side, then it is replaced with what is on the rule's right side.

Surface grammar allows the input text to be segmented into syntactic phrases. Since there is nothing that can decide whether a noun phrase is a good candidate to compound term or not, a simple filter can give us all the candidates to compound terms right after this process without any further process. Later on, if for a specific domain there are some specific better forms the surface grammar can be rewritten in order to reflect that prior knowledge. For now a generic tool is needed so this case is out of our scope.

Frequency of words and noun phrases computed over a general-content corpus [4] allows us to detect terms by comparing frequencies of the entities in the text being analysed and the ones in the corpus of reference.

ATA's output is divided into two sets both of which may be empty. The first set contains simple term candidates identified in the text. The second set contains compound terms candidates detected in the text.

4 PAsMo

PAsMo is a post-morphological analyzer rule-based rewriter whose function is to perform the last processing phase before the syntactic analysis. It is based on the MPS application developed at GRIL. Using recomposition and correspondence rules, it makes some processing based on the morphological characteristics of words rewriting word sequences and changing the tags used. In addition, it splits the text into segments.

PAsMo receives a text where all the words were previously enriched with is morphological features. The old version used Prolog and wasn't efficient enough considering that large corpora had to be analyzed.

Changing the chosen language to C++ and enhancing the algorithm, reduced processing time by a factor of 20 times on the best cases (and even more on worst cases, that is were ambiguity in sequences of words exists); XML input and output is now possible facilitating the communication between modules and data verification; Operators that allows to constrain the minimal number of times a words should appear so the rule can be applied, reduces the number of needed rules.

5 Term Extraction

This is the process responsible for the terms' detection. As the two types of terms require different processes.

Given the characteristics of simple terms, their processing starts by collecting all input words appearing with frequency higher than the threshold defined by the corpus. From this collection, words classified as nouns may be considered as simple term candidates. Those classified as unknown but considered as nouns due to the syntactic process are also considered as candidates.

Compound terms obey some syntactical and grammar restrictions that make the detection process easier. The structure of the word sequences that are to be considered as compound term candidates are described in an external file merged on the grammar used in the syntactic analysis.

In an hybrid system like this one, high frequency terms will be detected statistically; and low frequency terms (e.g. 1 or 2 utterances) will be detected through the grammar of terms. Then it will be necessary to review both term candidates (those given statistically and those given by the grammar). Yet, we believe that if a term candidate in specialised texts as low frequency that means that maybe they are not terms in the field under scope. In a specialised text of a given domain, terms from that domain will have high chances of being high frequency terms.

In addition, the system must distinguish between high-frequency words occurring within compound terms and isolated occurrences of the same words.

For instance, consider the expression "worldwide computer network". The word "network" will appear at least as much as the expression. So, after the process extracts the candidate terms the system must verify whether the word "network" is also a term, i.e., if the word appears enough times by itself. Yet, the compound has, theoretically, great possibility of being a hyponimic term (with a more restricted reference) and "network" to be it's correspondent generic term, presenting a broader reference [5].

"Computer network" will also be a term candidate (in fact it is a term of a hierarchically higher level). The grammar of terms is designed to give fewer chances to be term candidates to longer nominal phrases (with more components).

On one hand, we think there is no way (for the time being) of letting out sequences like those suggested. Yet, one the other hand, sequences like those presented are predictably low frequency sequences in specialised texts. If they are very frequent it means that we have to reconsider if they are (are not) actual terms of the domain under study, or let that checking for the linguistic using the system.

If one of the criteria when finding terms is the frequency of a word when comparing with general corpora, we need to extract frequencies for each lemma. For that the same architecture will be used enriched with a tool for resolve ambiguity that remains at the end of the process chain. The corpora we are going to use is CETEM-Público [15].

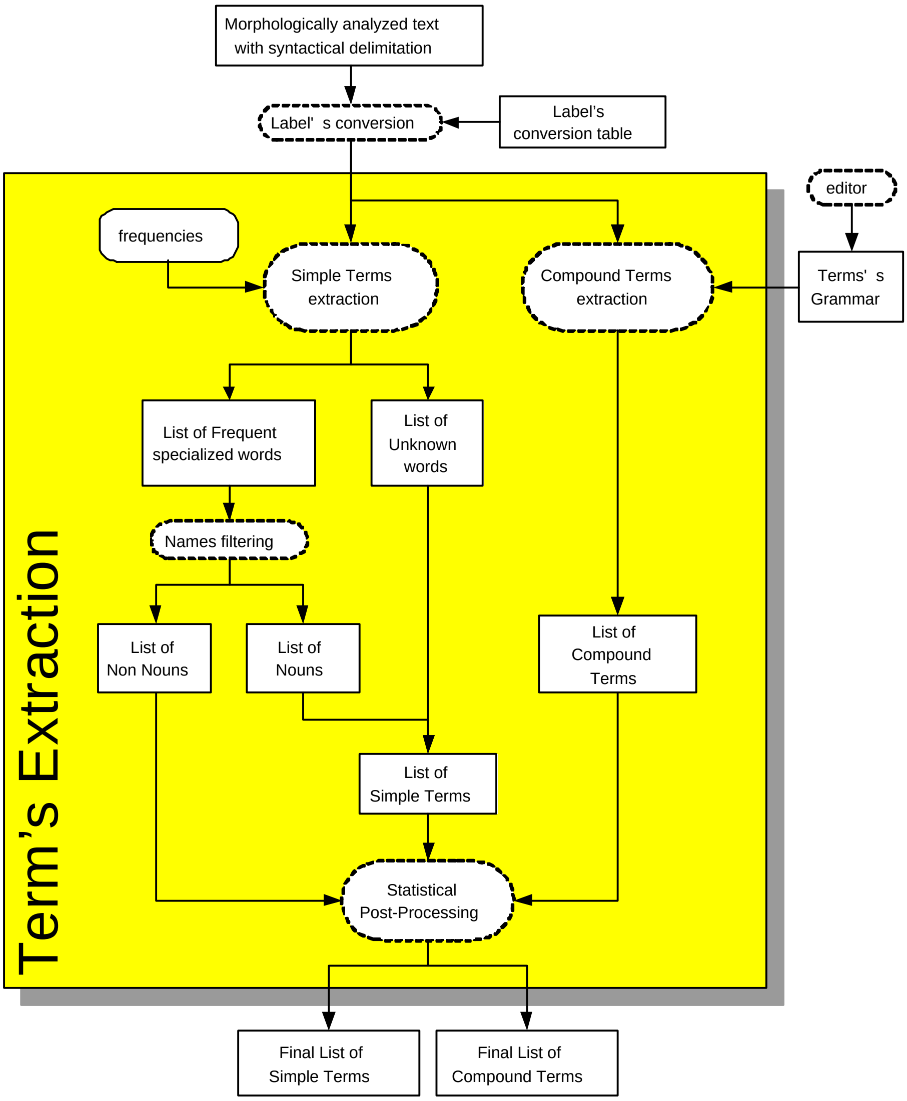

Fig. 2. ATA's Architecture.

6 Conclusions and Future Directions

In this article a new system for automatic term acquisition has been described. We are especially interested in studying the capability and the implications of building an automatic term acquisition system.

The ATA system will probably be a useful helper in solving the problem of the semi-automatic building of terminological indexes and will be used on different kinds of specialized documents.

We think that knowledge acquisition for knowledge-based systems is also a suitable experimentation ground for such a terminology extraction system, provided that an appropriate tool exists to represent and record information.

Although some of the processes are already implemented, at this time the system is not yet completely functional. That is, the final process, the one responsible for term extraction, has not been implemented yet. Currently, processing stops after syntactic analysis that means that all the relevant linguistic information was gathered and the statistical process can now begin. For now, the architecture and the main algorithm are defined so the implementation process can begin as soon as possible.

The system will be evaluated using the described methods (see 2), hopping to achieve results similar to those of systems for Portuguese [6, 18] and for other foreign languages as English [13] and French [7].

Probably, the price to pay for an automatic acquisition without any intermediary human validation is twofold: the procedure can let relevant information pass through undetected; it can acquire false information. This is the reason why perfecting these procedures requires the adoption of experimental processes, with numerous tests carried on large-scale corpora, that ensure the global empirical validity of the procedures.

Also as a test, we are going to use the system to create automatic subject indexes of specialized books and compare them with the original ones.

Changing language, currently, means having to restart the whole system. This entails considering the new language's dictionary, analysing the morphological behaviour of that language to write new rules, analysing a general corpus and computing word frequency and getting a surface grammar. That work can be done in order to prove that the system is language-independent.

Format information can also be considered by the system. The text format depends on the editor. It is necessary to create an external format description. If the text format has been lost at some point, the initial text has to be considered.

The idea is to give more or less importance to a word according to its format [11]. For instance, if a word appears in a title it must be considered more important that a word that is in the middle of a paragraph. Thus, it is possible to create a hierarchical classification that considers input text format. This classification should consider: Titles of documents, sections, and subsections; Bold, italic, and underline; Type and size of letter; Caps usage; Headers and footers; Footnotes; Quotations; Other styles.

References

1. Salah Aït-Mokhtar. *L'analyse Présyntaxique en une seule étape.* PhD thesis, Université Blaise Pascal, Feb 1998.
2. Fernando Batista. Análise sintáctica de superfície e consistência de regras. Master's thesis, Instituto Superior Técnico, UTL, 2002. (work in progress).
3. D. Bourigault. Surface grammatical analysis for the extraction of terminological noun phrases. *Proceedings of the 15th International Conference on Computational Linguistics, COLING'92*, 1992. p. 977-981.
4. P.R. Clarkson and R. Rosenfeld. Statistical language modeling using the cmu-cambridge toolkit. *Proceedings ESCA Eurospeech*, 1997.
5. D. A. Cruse. Lexical semantics, 1986.
6. J. Ferreira da Silva and G. Pereira Lopes. A local maxima method and a fair dispersion normalization for extracting multi-words units from corpora. *International Conference on Mathematics of Language, Orlando*, July 1999.
7. B. Daille. Study and implementation of combined techniques for automatic extraction of terminology. *The balancing act combining symbolic and statistical approaches to language*, pages 49–66, 1996.
8. Rosa Estopà. *Les unitats terminològiques polilexemàtiques en els lèxics especialitzats: dret i medicina.* PhD thesis, Institut Universitari de Lingüística Aplicada, Barcelona, UPF, 1999.
9. Abbaci Faiza. Développement du module post-smorph. Master's thesis, Mémoire de DEA de linguistique et informatique, GRIL, Université Blaise Pascal, Clermont-Ferrand, 1999.
10. Caroline Hagège. *Analyse syntaxique automatique du portugais.* Thèse de doctorat, Université Blaise Pascal, GRIL, Clermont-Ferrand, 2000.
11. C. Jacquemin. Quelques exemples d'application du traitement automatique des langues en accès à l'information. *5emes Journées Internationales d'Analyse de Données Textuelles (JADT)*, 1, 2000.
12. C. Jacquemin and D. Bourigault. Term extraction and automatic indexin. *R. Mitkov, editor, Handbook of Computational Linguistics*, 2000.
13. J. S. Justeson and S. M. Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering, p. 9-27*, 1995.
14. C. D. Manning and H. Shutze. *Foundations of Statistical Natural Language Processing.* MIT Press, London, 1999.
15. MCT and Público. Cetempúblico - corpus de extractos de textos electrónicos, 2000.
16. A. P. Marquez Neto. Terminologia e corpus linguístico. *Revista Internacional de Língua Portuguesa - RILP n. 15, p. 100-108*, 1996.
17. Joana Lúcio Paulo. Pasmo - pós-análise morfológica. Relatório técnico, Instituto Superior Técnico, Lisboa, 2001.
18. J. Silva, G. Dias, S. Guilloré, and G. Lopes. Using localmaxs algorithm for the extraction of contiguous and non-contiguous multiword lexical units. *9th Portuguese Conference on Artificial Intelligence*, 1695:113–132, September 1999.