# Extraction of the complex terms: the contribution of categorial grammars

Ismaïl Biskri [(*)] [(**)] ; Jean-Guy Meunier [(**)] ; Sylvain Joyal [(*)] ; Frédéric Gayton [(*)]

[(*)] Département de Mathématiques et Informatique
Université du Québec à Trois-Rivières
Trois-Rivières (QC) Canada, G9A 5H7
1 819 376 5011 #3837
biskri@uqtr.ca

[(**)] Laboratoire d'Analyse Cognitive de l'Information
Université du Québec À Montréal
Montréal (QC) Canada, H3C 3P8
1 514 987 3000 #0339
meunier.jean-guy@uqam.ca

## ABSTRACT

A long time categorial grammars were regarded as "toys grammars ". Indeed, in spite of a very solid theoretical base, categorial grammars remain rather marginal as soon as it is a question of conceiving concrete applications. However, this model of grammars has an unquestionable advantage compared to the majority of the other grammatical models: it is multilingual; multilingualism becoming, with the rise of the Web, one of the most significant constraints in the development of tools for natural language processing. In our article we propose a multilingual approach for the extraction of the complex terms using a linguistic filter founded on a categorial model. This filter belongs to a set of filters, some being of linguistic type, others of statistical type. We will describe our filters, however we will insist on that built by means of categorial grammars. Finally, our approach is by design interactive and constantly under the user's control.

**Keywords**
Complex terms extraction; multilingual approach; categorical grammars

## 1. INTRODUCTION

The complete and accurate identification of terms in a specific domain or corpus is considered as a pre-processing of the highest importance for the production of adequate and reliable results in various applications, such as information retrieval, indexation, written or spoken language processing, translation, summarisation, information or document management, of course terminology, and in the last decade ontology [14]. In recent years, a number of tools dealing with terms have been developed and proposed in the literature. These tools typically accept on input a text or corpus, either pre-processed (e.g. tagged) or not and automatically produce a list of candidate terms, often via statistical (Bayesian) computations or linguistic one[11] [5] [7] [9] [12]. Statistical approaches can be multilingual, but they are however noisy [14] [11], and terms frequencies are sometimes false, especially when there is not lemmatisation [8]. Linguistic approaches are less noisy, but however they can't deal with multilingual corpora or certain neologisms in specific domains. These approaches seem adapted to well stereotyped texts [4] [6] [8] [10] [12].

The method and tool we present in this paper also allows the extraction of recurring expressions (i.e. complex terms), from a corpus. However, our approach is considerably different from others:

- By design, it is interactive and constantly under the user's control. The glance of the user (expert) is significant. Different users using the same software can end up with different results. The same complex terms are not necessarily similar, for example, in medicine and anthropology.
- While combining statistical filters and linguistic filters, our approach tends to be multilingual.
- The software has learning capabilities. Complex terms identification processing is based on previous complex terms instances already validated by the user. It clearly makes an interesting and useful addition from the user's viewpoint. This learning capability improves the performance of the software.

## 2. THE SOFTWARE'S MAIN PROCESSING PHASES

Our approach combines a basic statistical Bayesian computation with both numeric and linguistic filters. Most of our filters are computationally inexpensive to apply and easily amenable to the processing of other languages than French and English.

Our software, called ESATEC (for Extraction Semi-Automatique de Termes Complexes – Semi-Automatic Extraction of Complex Terms), accepts on input a raw textual corpus, which is neither tagged nor lemmatised, from which it extracts its lexicon. The only *a priori* information (language dependent but domain independent) we need is that contained in the some following lists: functional words list, verb list, adverb list, **categorial types dictionary** etc. Once the software has produced the lexicon contained in the corpus, the user selects words of interest to her. These could be specific words that can function as *term kernels* or even the whole lexicon (term kernels are words around which other words can appear to form a valid term. For example, in "acid sulphuric" and "acid hydrochloric", the word 'acid' is a kernel). The user also specifies the size in number of words of the complex terms. In a second phase, the software computes the N-grams of words from the corpus, using the information specified by the user in the initial phase. At this point, all we have are N-grams of words, some of which will correspond to candidate terms. Simultaneously, the software constructs a collocation matrix that will be used to compute the candidate terms probabilities, based on Bayes' generalised rule. The Bayesian computation determines the probability of sequences of words within the input corpus. The highest the probability of a particular N-gram, the more the user will tend to conclude that this n-gram of words corresponds to a term. In this sense, the Bayesian probability acts as an indicator for the user to decide whether a candidate term should be considered as a legal term or not. But these probabilities can be said to represent an approximation to a complex linguistic phenomenon. In particular, they tend to contain a certain amount of noise (i.e. low precision) which makes the user's decision process more difficult and more time consuming. In order to get rid of false candidates, the software then applies a number of semi-automatic numeric and linguistic filters. These filters are independent and the order in which they are applied is up to the user. As we have shown elsewhere [2], a hybrid combination of statistical and linguistic models can positively influence the efficiency of pure numerical methods by improving the granularity of outputs and, thus, their utility value for the user. The same productive idea is used here: the combination of a Bayesian computation with simple and flexible numeric and linguistic filters allows us to eliminate a lot of noise produced by the basic Bayesian model applied here. A numeric computation will also be applied during the learning phase in order to detect term structures already encountered and validated by the user.

Let us now provide a brief description of our three first filters and a more detailed of the fourth one. The order of presentation of the filters is not significant.

The first filter eliminates candidate terms that have a probability lower than a certain threshold. This threshold is set by the user who can experiment at ease with it. It could also be the case that certain thresholds will have been established after lots of testing.

The second filter eliminates candidate terms that begin or end with functional words (such as determiners, conjunctions, etc.) or verbs, adverbs, etc. This is where the domain-independent *a priori* information we mentioned is needed. Of course the user can avoid all or a part of this filter. This kind of filter has been used previously in Lexter project [4]. It makes it possible to determine the border of the candidate term.

The third filter eliminates candidates that begin or end with specific words identified by the user in the lexicon. In that case, it is the user's knowledge of the corpus domain that can be useful in identifying non-productive words. The software contains no domain-specific *a priori* knowledge.

The fourth filter, by applying a syntactic analysis, eliminates candidate terms who are not nominal groups. Similar filters have been shown elsewhere, especially in [8]. What characterizes them is their strong dependence to a given language. Our linguistic filter is founded on the model of Applicative and Combinatory Categorial Grammar [3]. It is a universal model, whose "the spinal column" is independent of the language. Several works show the relevance of categorial grammars in the syntactic analysis of statements of different languages (French, English, Dutch, Arabic, etc.) [3] [13]. Problems were of course identified, mainly, with complex constructions like coordination, subordination, etc. Our concern not being the study of these constructions, it is thus possible to simplify the model. Applicative and Combinatory Categorial Grammar conceptualizes the language as a succession of linguistic units of which some function like operators whereas others function like operands. This is represented by the assignment of syntactical categories, which are provided in a dictionary, to each linguistic unit. Syntactical categories are orientated types developed from basic types (for instance N (nominal group) and from two constructive operators '/' and '\' (if X and Y are orientated types then X/Y and X\Y are orientated types[1]).
Following to this assignment an inferential calculation on the categories is applied, by means of categorial rules, to check the nominal aspect of the candidate terms. Let us provide a sample of these rules[2] :

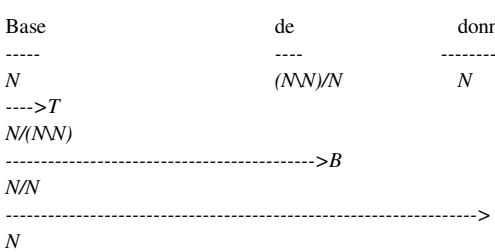
| | | |
|---|---|---|
| Applicative rules : | X/Y - Y → X | (**>**) |
| | Y – X\Y → X | (**<**) |
| Type-raising rules : | X → Y/(Y\X | (**>T**) |
| Functional composition rules : | X/Y - Y/Z → X/Z | (**>B**) |

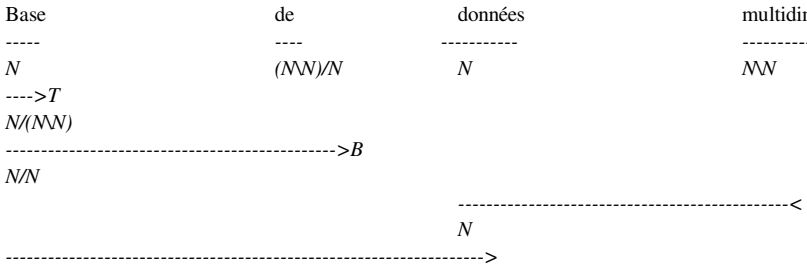
With this last filter, a full processing based upon Applicative and Combinatory Categorial Grammar is carried out in two main steps:

(i) The first step is illustrated by the assignment of categories to the linguistic units.

(ii) The second step is illustrated by the checking of the proper syntactic connection. In other words, here is checked the nominal syntagm nature of the candidate term.

For instance let us consider the inferential calculation of the following candidate terms (in french): (i) données fausses (false data); (ii) base de données (data base); (iii) base de données relationnelle (relational data base); (iv) fondement de la théorie des nombres


[1]X/Y and X\Y are functional orientated types. A linguistic unit with the type X/Y (respectively X\Y) is considered as operator (or function) whose typed operand Y is positioned on the right (respectively on the left) of operator.

[2] The rules, here, were simplified. For a thorough reading of the model, the reader might have to consult [4].

*N*

---

Fondement de la théorie des nombres

---

*N* *(N\N)/N* *N/N* *N* *(N\N)/N* *N*

---->*T*

*N/(N\N)*

---------->*B*

*N/N*

---------->*B*

*N/N*

---->*T*

*N/(N\N)*

---------->*B*

*N/N*

---------->

*N*

---------->

*N*

All these candidate terms are of category N. It is that which the filter need to validate them. They follow certain French patterns described in [8] [12]: Noun Adjective (i); Noun "de" (Determiner) Noun (ii); Noun "de" (Determiner) Noun Adjective (iii); Noun "de" (Determiner) "la" (Determiner) Noun "des" (Determiner) Noun (iv). Of course these patterns are not common for all the languages. However, Applicative Combinatory Categorial Grammar is suitable for other languages. If we have to process for example English, we have just to get a dictionary for English categories. We keep the same categorial rules.

## 3. LEARNING

The filters, presented in the preceding section, allow us to eliminate noise in the raw list of candidate terms since this first list is based exclusively on N-grams of words computations. Getting rid of noise improves precision. However, this improvement often leads to a lower recall value. In order to deal with this constraint, noise filtering is subject to a further requirement: *candidate terms cannot be eliminated by a filter if they have previously been approved by the user as valid terms*. In other words, terms learned by the system as legal terms are used in an ultimate term filtering verification. Clearly, the more text the software will process, the more extensive its list of learned terms will get and, consequently, the more discriminating this verification will get. What is nice about this ultimate rule is that it improves recall without affecting precision. But there is more: *candidate terms cannot be eliminated by a filter if they derive from valid terms*. For instance, if the list of learned terms contains "acid sulphuric" and if we are currently considering "acid hydrochloric" as a potential candidate term, then a simple function could ensure that such valid candidates are not eliminated. Indeed, a test stipulating that valid are the terms whose n-grams of characters decomposition is similar to a learned term one according to a certain threshold. We define N-gram of characters as a sequence of N characters. For instance if we take N = 3 the n-grams of characters decomposition of the first term "acid sulphuric" would be: aci, cid, id , d s, su, sul, ulp, lph, phu, hur, uri, ric ; the n-grams of characters decomposition of the second term "acid hydrochloric" would be: aci, cid, id , d h, hy, ydr, dro, roc, och, chl, hlo, lor, ori, ric. The two complex terms are regarded as similar if they share a certain number of trigrams, this number being higher than a certain threshold set by the user [1].

## 4. EVALUATION

Because our system is semi-automatic, comparing it to others is awkward, so a word of caution is in order at the beginning of this evaluation section. We felt we had to look at evaluation from a slightly different perspective. Our perspective is more qualitative evaluation than quantitative one.

For our evaluation, we took an online book: Out of Control by Kevin Kelly (http://www.well.com/user/kk/OutOfControl/index.html). The document is about 90 pages (54 147 words / 606 words per page). The corpus here is in English. Without reading the text, we processed this corpus with the software we have described above and we obtained a total population of candidate terms equal to 37423. The results shown in Figure 3 below presents the 20 first (two words) complex terms obtained, each separated by '#'.

**FROM THE ONLINE BOOK: OUT OF CONTROL BY KEVIN KELLY**

hive mind # complex systems # feedback loop # swarm systems # von Neumann # world war # nonzero sum # th century # gun barrel # fast cheap # artificial intelligence # zero sum # automatic control # living organisms # self control # steam engine # mark Pauline # san Francisco # mirrored box # stuart pimm.

Of course, given the semi-automatic nature of our software, this list reflects at least in part our own (naive) vision of the evaluation corpus considered here. Domain experts using our software would certainly have come up with (at least partially) different results. And this is fine! In fact, this is exactly why we argue that term identification is a user-specific task that should be supported by a flexible

semi-automatic tool rather than a purely automatic one. On another hand, we do not present any precision and recall figures as they would be meaningless, again because the approach is semi-automatic and its results are influenced by the user's subjectivity and (lack of) knowledge of the corpus' domain. Another important aspect is the user's goals when performing her task of term identification: is she looking for "standard" terms for indexing purposes, or for terms that would summarise the contents of the corpus, or for new terms that she, as an expert, would not already know? The software we presented above is able to deal, within certain limits, with this customisation dimension.

## 5. CONCLUSION

We have presented in our paper a linguistic filter integrated to a semi-automatic software tool for complex term identification. Our filter is different from most other term identification linguistic filters in that :

- It tends to be multilingual. With the growth of the Web and of the multilingual textual data bases, this aspect is significant.
- Even if our filters eliminate all candidate terms who are not nominal syntagm, through the phase of learning, our tool accepts other complex terms not of nominal group category. This is important, because the syntactic structure of the term is generally of the nominal group but this rule suffers from the exceptions (for example proper names: Abraham Lincoln) and is not true for all the domains.

Finally, we justify all our choices by the need to have a more flexible and customisable tool to perform, in a relatively reasonable time, term identification tasks for different languages. More specifically, in certain situations we want to allow the user's perspective, knowledge and subjectivity, influence the results.

## 6. REFERENCES


[1] Biskri, I. & Delisle, S. (1999). "Text Classification and Multilinguism: Getting at Words via N-grams of Characters" *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI-2002)*, Orlando (Florida, USA), 14-18 juillet 2002, Volume V, 110-115.

[2] Biskri, I. & Delisle, S. (1999). "Un modèle hybride pour le textual data mining - un mariage de raison entre le numérique le linguistique", *Actes de la 6ème Conférence Annuelle sur le Traitement Automatique des Langues Naturelles (TALN-99)*, Cargèse (Corse), France, 12-17 juillet 1999, 55--64.

[3] Biskri, I., Descles, J.P., (1997), "Applicative and Combinatory Categorial Grammar (from syntax to functional semantics)", in Recent Advances in Natural Language Processing (selected Papers of RANLP 95) Ed. Ruslan Mitkov & Nicolas Nicolov. John Benjamins Publishing Company, Numéro 136, pages 71-84.

[4] Bourigault, D. (1996). "Conception et exploitation d'un logiciel de termes : problèmes théoriques et méthodologiques", *IVème journées scientifiques du réseau thématique_— AUPELF-UREF*, Lyon, France, 1996, 137--146.

[5] Collier, N., Hirakawa, H. & Kumano, A. (1998). "Machine Translation vs Dictionary Term Translation – a Comparison for English-Japanese News Article Alignment", *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98)*, Montreal (Quebec), Canada, 263--267.

[6] Condamines, A., Rebeyrolle, J**.** (2001), "Searching for and identifying conceptual relationships via a corpus-based approach to Terminological Knowledge Base (CTKB)", in D. Bourigault, C. Jacquemin & M.-C. L'Homme (eds), Recent Advances in Computational Terminology, Amsterdam/Philadelphia, John Benjamins Publishing Company, pp. 128-148.

[7] Dagan I., Church, K. (1994). "Termight : Identifying and Translating Technical Terminology", *Proceeding of the Fourth Conference on Applied Natural Language Processing*, Association for Computational Linguistics, Stuttgart, Germany, 13-15 October 1994, 34--40.

[8] Daille, B. (1994). "Study and Implementation of Combined Techniques for Automatic Extraction of Terminology", *Proceedings of the Combining Symbolic and Statistical Approaches to Language Workshop (the Balancing Act)*, Las Cruces (New Mexico), USA, 1st July 1994, 29--36.

[9] Frantzi, K.T. (1997). "Incorporating context information for the extraction of terms", *Proceeding of the 35th Annual Meeting and the 8th Conference of the European Chapter of the Association for Computational Linguistics*, Madrid, Spain, 7-12 July 1997, 501--503.

[10] Frath P., Oueslati R., Rousselot F. (2000), "Identification de relations sémantiques par repérage et analyse de cooccurrences de signes linguistiques ". In *Ingénierie des connaissances. Evolutions récentes et nouveaux défis.* Eds. Jean Charlet, Manuel Zacklad, Gilles Kassel, Didier Bourigault. Eyrolles, Paris.

[11] Smadja, F. (1993). "Retrieving collocations from text: Xtract". *Computational Linguistics*, 19(1):143-178

[12] Sta, J.D. (1998), "Automatic acquisition of terminological relations from a corpus for query expansion". Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, p.371-372, August 24-28, 1998, Melbourne, Australia

[13] Steedman, M. (2000). The Syntactic Process, MIT Press/Bradford Books.

[14] Strzalkowski, T. (1999). Ed., *Natural Language Information Retrieval*, Kluwer Academic Publishers.