Annotation of the complex terms in multilingual corpora

Ismaïl Biskri, Boubaker Hamrouni & Nicole Munyana

LAboratoire de Mathématiques et Informatique Appliquées
Département de Mathématiques & Informatique, Université du Québec à Trois-Rivières
CP500 Trois-Rivières, Québec, Canada, G9A 5H7
Ismail.Biskri@uqtr.ca

ABSTRACT

For 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 show a multilingual approach for the extraction of the complex terms using a linguistic filter founded on a categorial model.

Introduction

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, 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 recent years, a number of tools dealing with terms have been developed and proposed in the scientific literature (Dagan & Church, 1994) (Condamines, Rebeyrolle, 2001). 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. Statistical approaches can be multilingual, but they are however noisy. 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 (Strzalkowski, 1999).

Multilingualism becoming, with the rise of the Web, one of the most significant constraints in the development of tools for natural language processing, it consequently becomes important to consider approaches which take into accounts this aspect. Precisely, our general approach has capabilities to be multilingual (Biskri & al., 2004). The method we use is a hybrid method. It combines a basic statistical Bayesian computation (without any smoothing technique) 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, the latter being the only language considered in this paper.

Our input text is a simple text file which is neither tagged nor lemmatised. The only *a priori* information we need is that contained in the some following lists: functional words list, verb list, and adverb list, etc. or a categorial dictionary —this *a priori* information is language dependent but domain independent. The Bayesian computation determines the probability of (ordered) sequences of words within the input corpus. The highest the probability of a particular N-gram of words, 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. As we have shown in (Biskri & al., 2004), a hybrid combination of statistical and basic linguistic filters improves the granularity of outputs even if there remains a residual noise which can appear significant in certain cases. It is to reduce this residue that a filter based on categorial grammars was added with the data processing sequence. This filter can have a dual use. Either, it makes it possible to remove candidate terms of which the grammatical category is not that of the nominal group, or, it makes it possible to

preserve candidate terms who could be removed by the other filters. The input of this filter is a list of candidate terms.

We present in what follows the theoretical details of this filter.

The model of Applicative and Combinatory Categorial Grammar

The model of Applicative and Combinatory Categorial Grammar (ACCG) falls under a paradigm of language analysis that allows a complete abstraction of grammatical structure from its linear representation due to the linearity of the linguistic signs and a complete abstraction of grammar from the lexicon. According to the framework of Applicative and Cognitive Grammar (Desclés 1990, 1996) and Applicative Universal Grammar (Shaumyan 1998), the language analysis has to postulate three levels of representation: Phenotype, Genotype and cognitive levels (in this paper we are interested only in the two first levels).

In the *phenotype level*, particular characteristics of natural languages are expressed (for example order of words, morphological cases, etc...). The linguistic expressions of this level are concatenated linguistic units according to the syntagmatic rules of the language concerned. We will write them as follows: lets u1, u2, u3 linguistic units, their concatenation representation in phenotype is u1 – u2 – u3.
In the *genotype level*, grammatical invariants and structures that are underlying to sentences of phenotype level are expressed. The genotype level uses a variable-free formal language, called *Genotype Calculus*, as its formal framework. In this level functional semantic interpretations are expressed by means of combinators, which are abstract operators who allow constructing more complex operators. According to (Curry and Feys 1958) each combinator is associated with to a β-reduction rule. For instance, we present combinators **B**, **C**, **C***, with the following rules (U_1, U_2, U_3 are typed applicative expressions) :

(**B** U_1 U_2) U_3) –> (U_1 (U_2 U_3))
(((**C** U_1) U_2) U_3) –> ((U_1 U_2) U_3)
((**C*** U_1) U_2) –> (U_2 U_1)

Applicative and Combinatory Categorial Grammar (ACCG), (Biskri and Desclés 1997), explicitly connects phenotype expressions to its underlain representations in the genotype (functional semantic interpretation). It, like all Categorial Grammar models (Morrill 1994) (Moorgat 1997) (Steedman 2000) (Dowty 2000), assigns syntactical categories to each linguistic unit. Syntactical categories are orientated types developed from basic types and from two constructive operators '/' and '\'. A linguistic unit 'u' with the functional 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 and the result is of type X. In our paper, a linguistic unit u with orientated type X will be designed by '[X : u]'.

Let us provide now ACCG rules used in this paper.

Application rules :	[X/Y : u_1] - [Y : u_2] ----------> [X : (u_1 u_2)]	[Y : u_1] - [X\Y : u_2] ----------< [X : (u_2 u_1)]
Permutation rules :	[(X\Y)/Z : u] ---------->**C** [(X/Z)\Y : (**C** u)]	
Functional composition rules:	[X/Y : u_1]-[Y/Z : u_2] ---------->**B** [X/Z : (**B** u_1 u_2)]	

The premises in each rule are concatenations of linguistic units with orientated types considered as being operators or operands, the consequence of each rule is an applicative typed expression with an eventual introduction of one combinator. The permutation of a unit u introduces the combinator **C**; the composition of two concatened units introduces the combinator **B**.

Since the aim of the approach we present here is to find in a corpora the complex terms (nominal groups), it is not useful to consider the type raising rule in the theoretical formalism. This rule, initially, used to give an account for the cases of coordination with ellipse (Steedman, 2000) (Dowty, 2000), requires a whole set of meta-rules to control its release (Biskri, Desclés, 1997). This rule makes it possible, also, to choose a strategy of

incremental analysis “from left to right" (Biskri, Desclés, 1997). Our concern, here, is not to analyze coordination. However, it appears significant to us to preserve an analysis “from left to right" for reasons which we will explain further. With this intention, the use of the permutation rule is relevant. Indeed, this rule does not require any meta-rule for its release on the one hand. On the other hand, it allows a coherent analysis with a strategy “from left to right”. In addition, from a technical point of view, the combinator **C,** which is introduced in the syntagmatic expression by the permutation rule, may be equivalent to a combination of combinators **C*** and **B**, respectively introduced in the syntagmatic expression by type raising and composition rules. In fact, the following combinatory expressions (a) and (b) are equivalent. according to the theorem of Church-Rosser.

a) (((**C** X) Y) Z)
b) (((**B** (**C*** Y)) X) Z)

Indeed,

The β-reduction process of (a) is

(((**C** X) Y) Z)
((X Z) Y)

The β-reduction process of (b) is

(((**B** (**C*** Y)) X) Z)
((C* Y) (X Z))
((X Z) Y)

The normal form of (a) is the same as the normal form of (b). According to the theorem of Church-Rosser the combinatory expressions (a) and (b) are then equivalent.

A full processing based upon Applicative and Combinatory Categorial Grammar is carried out in three 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 phrase nature of the candidate term.
(iii) The third step is illustrated by the constructing of the normal form.

For instance let us consider the inferential calculation of the following candidate terms (in french):

(i) *Base fondamentale* (fundamental base);
(ii) *Base de données* (data base);
(iii) *Base de données relationnelle* (relational data base);
(iv) *Fondement de la théorie des nombres* (base of the theory of the numbers)

Example 1 :

1. [NP: *base*] - [NP\NP: *fondamentale*]
2. [NP: (*fondamentale base*)] (<)

3. (*fondamentale base*)

Example 2 :

1. [NP: *Base*] – [(NP\NP)/N: *de*] – [N: *données*]
2. [NP: *Base*] – [(NP/N)\NP: (**C** *de*)] – [N: *données*] (>**C**)
3. [(NP/N) : ((**C** *de*) *Base*)] – [N: *données*] (<)
4. [NP: (((**C** *de*) *Base*) *données*)] (>)

5. (((**C** *de*) *Base*) *données*)
6. ((*de données*) *base*) **C**

Example 3 :

1. [NP: *Base*] – [(NP\NP)/N: *de*] – [N: *données*] – [N\N: *multidimensionnelles*]
2. [NP: *Base*] – [(NP/N)\NP: (**C** *de*)] – [N: *données*] – [N\N: *multidimensionnelles*] (>**C**)
3. [(NP/N) : ((**C** *de*) *Base*)] – [N: *données*] – [N\N: *multidimensionnelles*] (<)
4. [NP: ((**C** *de*) *Base*) *données*)] – [N\N: *multidimensionnelles*] (>)

5. [(NP/N) : ((**C** *de*) *Base*)] – [N: *données*] – [N\N: *multidimensionnelles*] (**Structural Reorganisation**)
6. [(NP/N) : ((**C** *de*) *Base*)] – [N: (*multidimensionnelles données*)] (<)
7. [NP: (((**C** *de*) *Base*) (*multidimensionnelles données*))] (>)

8. (((**C** *de*) *Base*) (*multidimensionnelles données*))
9. ((*de* (*multidimensionnelles données*)) *Base*) **C**

Example 4 :

1. [NP: *Base*] – [(NP\NP)/NP: *de*] – [NP/N: *la*] – [N: *théorie*] – [(N\N)/N: *des*] – [N: *nombres*]
2. [NP: *Base*] – [(NP/NP)\NP: (**C** *de*)] – [NP/N: *la*] – [N: *théorie*] – [(N\N)/N: *des*] – [N: *nombres*] (>**C**)
3. [(NP/NP) : ((**C** *de*) *Base*)] – [NP/N: *la*] – [N: *théorie*] – [(N\N)/N: *des*] – [N*: nombres*] (<)
4. [(NP/N) : (**B** ((**C** *de*) *Base*) *la*)] – [N: *théorie*] – [(N\N)/N: *des*] – [N: *nombres*] (>**B**)
5. [NP: ((**B** ((**C** *de*) *Base*) *la*) *théorie*)] – [(N\N)/N: *des*] – [N: *nombres*] (>)
6. [(NP/N) : (**B** ((**C** *de*) *Base*) *la*)] – [N: *théorie*] – [(N\N)/N: *des*] – [N: *nombres*] (**Structural Reorganisation**)
7. [(NP/N) : (**B** ((**C** *de*) *Base*) *la*)] – [N: *théorie*] – [(N/N)\N: (**C** *des*)] – [N: *nombres*] (>**C**)
8. [(NP/N) : (**B** ((**C** *de*) *Base*) *la*)] – [(N/N) : ((**C** *des*) *théorie*)] – [N: *nombres*] (<)
9. [(NP/N) : (**B** (**B** ((**C** *de*) *Base*) *la*) ((**C** *des*) *théorie*))] – [N: *nombres*] (>**B**)
10. [NP: ((**B** (**B** ((**C** *de*) *Base*) *la*) ((**C** *des*) *théorie*)) *nombres*)] (>)

11. ((**B** (**B** ((**C** *de*) *Base*) *la*) ((**C** *des*) *théorie*)) *nombres*)
12. ((**B** ((**C** *de*) *Base*) *la*) (((**C** *des*) *théorie*) *nombres*)) **B**
13. (((**C** *de*) *Base*) (*la* (((**C** *des*) *théorie*) *nombres*))) **B**
14. ((*de* (*la* (((**C** *des*) *théorie*) *nombres*))) *Base*) **C**
15. ((*de* (*la* ((*des nombres*) *théorie*))) *Base*) **C**

All these candidate terms are of category NP. It is that which the filter need to validate them. They follow certain French patterns described in (Daille, 1994) (Sta, 1998):

(i) Noun Adjective (example 1);
(ii) Noun “de” (Determiner) Noun (Example 2);
(iii) Noun “de” (Determiner) Noun Adjective (example 3);
(iv) Noun “de” (Determiner) “la” (Determiner) Noun “des” (Determiner) Noun (example 4).

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.

The analyses shown here are “from left to right”. This kind of analyses eliminates the phenomenon of the pseudo-ambiguity which consists in building several trees of syntactic derivation which correspond to only one semantic interpretation.

For the two first examples, no spurious constituent is constructed. For the first example steps 1 and 2 are applied in phenotype level whereas the step 3 is applied in the genotype level. The expression obtained in the step 3 represents the functional form of the validated complex term. For the second example the step 1 assigns categorial types to linguistic units. Because the type of *Base* cannot be composed with the type of *de*, this last one undergoes in the step 2, an operation of permutation which introduces the combinator **C.** Steps 3 and 4 respectively operate the backward and the forward application rules.
The syntactic analysis “from left to right” raises the problem of non-determinism introduced by the presence in the language of backward modifiers that stand as operators which are applied to the whole or a part of a structure previously constructed. If, in the two first cases the use of application and permutation rule allows the analysis to be carried on, it is quite different for the third and the fourth examples where the analyses “block”.
For a term like *Base de données multidimensionnelles*, the parser at first creates the “spurious constituent” *Base de données* (according to the meaning of the sentence). This last constituent is not combinable with *multidimensionnelles*, since the type of *multidimensionnelles* is N\N whereas the type of *base de données* is N and no categorial rule can consequently be applied. As a matter of fact, *multidimensionnelles* is an operator whose operand *données* stands on its left. A quasi-incremental analysis “from left to right” makes easy the application of a combinatory categorial rule as soon as possible. This factor gets as direct consequence to “absorb” *données* into ((**C** *de*) *Base*) *données*), which obviously does not allow us to directly construct (*multidimensionnelles données*). That is to say, *données*

does not appear clearly as the operand of the operator *multidimensionnelles*.

The same problem is observed with example 4 at step 5.

The raised problem comes back to the possibility of a backtracking. But this backtracking is the kind one to increase the "computational" cost (memory and time execution) of one syntactic analysis. However, an "intelligent" backtracking (that we will propose later on) can allow us to reduce this cost considerably, and at the same time by constructing proper semantic analyses and by eliminating pseudo-ambiguities. So, such a backtracking will decompose the constituent already constructed in two components whose one of them may be combined with the backward modifier.

Formally, this operation of structural reorganization is realized by the two following successive steps:

a- As shown elsewhere (Biskri, Desclés, 1997) the reorganization of constituent already constructed isolates two sub-categories at each time, and tests if the backward modifier may "be combined on left" or not with one of these two sub-categories. We then proceed to the reduction of combinators until the test gives us a positive value. At the end of the process we will recover a new typed applicative structure "equivalent" to the first one.

Example: In the case of the statement *Base de données multidimensionnelle*, the steps of reorganization are :

Constituent constructed:
[NP : ((**C** *de*) *Base*) *données*)]

The two sub-categories are :
[(NP/N) : ((**C** *de*) *Base*)] ; [N : *données*]

données can be composed with *multidimensionnelles*.

Hence, no need in this case to combinators reduction process. We recover the category in output:
[NP : ((**C** *de*) *Base*) *données*)].

b- Decomposition realized by means of the two rules:

$$\frac{[X : (u_1\ u_2)]}{[X/Y : u_1]\text{-}[Y : u_2]}\ \textbf{>dec} \quad ; \quad \frac{[X : (u_1\ u_2)]}{[Y : u_2]\text{-}[X\backslash Y : u_1]}\ \textbf{<dec}$$

We read these rules as following:

- For (**>dec**): If we have an applicative structure (u_1 u_2) with type X, u_1 of type X/Y and u_2 of type Y, then we can construct a new concatened expression formed by both categories [X/Y:u_1] and [Y:u_2].

- For (**<dec**): If we have an applicative structure (u_1 u_2) with type X, u_1 of type X\Y and u_2 of type Y, then we can construct a new concatened expression formed by both categories [Y:u_2] and [X\Y:u_1].

Let us notice that the two rules **(>dec)** and **(<dec)** are respectively inverse to the rules of functional application (>) and (<). Both rules allow us to construct again a new concatenated ordering of the structure operator/operand coming from the reorganization.

For the statement *Base de données multidimensionnelles* the decomposition is applied to the structure that arises from reorganization: [NP: ((**C** *de*) *Base*) *données*)].

With the rule (**>dec**), we produce the concatenated ordering: [(NP/N) : ((**C** *de*) *Base*)] - [N : *données*].

We observe the same process which occurs at step 6 of the analysis of the candidate term *base de la théorie des nombres*. Indeed, the constituent ((**B** ((**C** *de*) *Base*) *la*) *théorie*) is spurious since *théorie* is operand of the modifier *des nombres*.

The most significant in our results is embodied, not only in the validation of the complex terms but especially in their functional structure. This one makes it possible to construct the environment of use of a basic concept represented by a word. In the analysed examples the word *base* is seen modified by several expressions which act as an adjective and which determine the context of its use. We can, in this way, systematize, for instance, the extraction of relations which could generalize the concept of hyperonymy or hyponymy. With this intention, the management of a complete semantic graph becomes necessary. In such a graph nodes can represent words and edges can contain features to categorize the relations between the nodes.

Conclusion

We have presented in our paper a linguistic filter integrated to a semi-automatic software tool for complex terms identification. Our filter is different from most other terms 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. All what we need to adapt the approach to a new language is another dictionary of categorial types with the lexical entries of this language

- A solid logical and linguistic theory supports the approach. This theory is particularly flexible. In certain cases the complex terms can be of which the grammatical category of verbal phrase. It would be enough then to consider that the complex terms to validate have the categorial types specific to the verbal phrases.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

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.

Biskri, I., Meunier, J.G., Joyal, S., (2004), "L'extraction des termes complexes : une approche modulaire semi-automatique" Dans Le Poids des mots (Actes des 7èmes Journées Internationales d'Analyse Statistique des Données Textuelles, Louvain-La-Neuve, Belgique), Gérard Purnelle, Cédrick Fairon & Anne Dister (eds). Presses Universitaires de Louvain, Volume 1, pp 192-201, ISBN 2-930344-49-0.

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.

Curry, B. H., Feys, R., (1958). *Combinatory logic* , Vol. I, North-Holland.

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.

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.

Desclés, J.P., (1996). Cognitive and Applicative Grammar: an Overview. *in C. Martin Vide, ed. Lenguajes Naturales y Lenguajes Formales, XII*, Universitat Rovra i Virgili. , 29-60.

Desclés, J. P, (1990). *Langages applicatifs, langues naturelles et cognition*, Hermes, Paris.

Dowty, D., (2000), The Dual Analysis of Adjuncts/Complements in Categorial Grammar. *In Linguistics 17*.

Moorgat, M., (1997). Categorial Type Logics. *In Johan Van Benthem and Alice Ter Meulen eds*., Handbook of Logic and Language, 93-177. Amsterdam: North Holland.

Morrill, G., (1994), *Type-Logical Grammar*. Dordrecht: Kluwer.

Shaumyan, S. K., (1998). Two Paradigms Of Linguistics: The Semiotic Versus Non-Semiotic Paradigm. *In Web Journal of Formal, Computational and Cognitive Linguistics.*

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

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

Srzalkowski, T. (1999). Ed., *Natural Language Information Retrieval*, Kluwer Academic Publishers.