Didier Bourigault, Centre d'Analyse et de Mathématiques Sociales, Paris
Isabelle Gonzalez-Mullier & Cécile Gros, Electricité de France, Direction des Etudes et Recherches, Clamart

LEXTER, a Natural Language Processing Tool for Terminology Extraction

Abstract

LEXTER is a terminology extraction software. It performs a morpho-syntactical analysis of a corpus of French texts on any technical domain and yields a grammatical network of noun phrases which are likely to be terminological units. This network of candidate terms, together with the corpus it has been extracted from, is then passed on to an expert for a validation by the means of a terminological hypertext web. The basic principle of LEXTER is that of splitting by locating terminological noun phrases boundaries. Non supervised corpus-based learning procedures allow the system to acquire lexico-syntactical information and to solve the problem of adjectives and prepositional phrases attachment. LEXTER is used in several Electronic Document Management projects to build different kinds of terminological products.

1. Introduction

LEXTER is a Terminology Extraction Software. A corpus of French texts on any technical subject is fed in it. LEXTER performs a morpho-syntactical analysis of this corpus and yields a network of noun phrases which are likely to be terminological units, representing the concepts of the subject field. This network of candidate terms, together with the corpus it has been extracted from, is then passed on to an expert for a validation by the means of a terminological hypertext web.

The development of a noun phrases extractor was a very delicate task. We were subject to two antinomic constraints: robustness and accuracy.

- Robustness: LEXTER has been developed in an industrial context, which is the Research and Development Division of the French Electricity Board. Thus, from the beginning of the project, we had decided to focus upon a strongly restrictive criterion: to apply the system over a wide range of texts. The texts analyzed are unrestricted texts gathered in large corpora. We had then to choose a fast and well-proved method. The system had to be really do-

main-independent. It could use no semantic nor conceptual information a priori given.

- Accuracy: The noun phrases the system extracts are the candidate terms which will directly be proposed to the user who builds the terminology of a domain. If the ratio of nonsense phrases is too high, the system could easily be rejected as a whole by the user.

We straightway dismissed the methods that are only statistical being incapable of satisfying the accuracy constraint. In order to satisfy the robustness and accuracy constraints, we were led to develop some original techniques of Natural Language Processing. We first present the basic principle of LEXTER (section 2). We then describe the different module of the system, the Splitting module (section 3), the Parsing module (section 4) and the Structuring module (section 5), with a particular emphasis on the Corpus-Based Endogenous Learning procedures we have implemented in the system. We conclude by a brief presentation of some results about the use of LEXTER in Electronic Document Management projects (section 6).

2. Basic principle: splitting by boundaries locating

The idea at the basis of the conception of LEXTER is that of locating noun phrases boundaries. Rather than exploiting knowledge "in the positive" on the possible grammatical structures of complex terms, we on the contrary rely on knowledge "in the negative" on the grammatical configurations which are known not to be parts of terms. The basic principle is then to split the text by locating these potential boundaries, between which noun phrases likely to be occurrences of terms are isolated.

In order to perform that splitting, we have implemented some techniques of local syntactical analysis by surface pattern. The input data of the module in charge of the splitting (the Splitting module) are just morphological information associated with each word of the text: grammatical category (part of speech), morphological features (particularly gender and number), lemmatized form. This information is given by the Tagging module, a morphological analyzer which has been developed by the Gsi-Erli company in the framework of the Graal Esprit project. The Splitting module is described in the next section.

3. The Splitting module

3.1 Local syntactical analysis

The techniques of local syntactical analysis which have been implemented in LEXTER consist in locating the morpho-syntactical patterns which cannot be parts of terminological noun phrases and which then are likely to indicate noun phrases boundaries. Some of these patterns are simple as, for example, verb, pronoun, preposition + possessive article, etc. The splitting phase produces a series of text sequences, most often noun phrases. These noun phrases may well be candidate terms themselves, but more often than not, they contain sub-groups which are also candidate terms. That is why we refer to these noun phrases as "Maximal-Length Noun Phrases" (henceforth MLNP). A simple example of splitting is given on figure 1. (See next page).

This idea of splitting is used within some other Natural Language Processing systems (see for example (Grefenstette 1992)). But, in the case of LEXTER, the noun phrases which are isolated by splitting are not intermediary data, as for example descriptors intended to be used by an other automatic module in order to index or classify documents. The extracted noun phrases, and the eventual sub-groups constituting them, will be the candidate terms proposed to the user. This requires a great precision in splitting.

3.2 Corpus-Based Endogenous Learning procedures

To precisely and correctly process some problematic splitting cases (particularly the coordination, the attributive past participle and the definite article "*le*"), it appears that the system must have at its disposal, and exploit, syntactical information of sub-categorization. We illustrate that situation by example [1]. The splitting module performs a normal split at the "*à une*" sequence, which corresponds to a boundary pattern. For the constraint of local syntactical coherence to be verified, that split must be done together with the elimination of the "*sensible*" adjective, whose eliminated sequence ("*à une*") introduces a complement, so that the system does not retain the syntactically invalid group which is "*armoire de contrôle sensible*" (cf. [1']).

Input text

L'opérateur passe en recirculation directe sur puisard grâce à la vanne manuelle d'isolement d'enceinte qui est équipée du clapet de sécurité rapide.

Tagged text

L [det.] opérateur [noun] passe [verb] en [prep.] recirculation [noun] directe [adj.] sur [prep.] puisard [noun] grâce à [prep.] la [det.] vanne [noun] manuelle [adj.] d [prep.] isolement [noun.] d [prep.] enceinte [noun.] qui [rel. pro.] est [verb] équipée [past part.] du [prep.] clapet [noun] de [prep.] sécurité [noun] rapide [adj.] . [typo]

Splitted test

Maximal Length Noun Phrases	(Boundaries)
	(L')
opérateur	passe (en)
recirculation directe sur puisard	grâce à (la)
vanne manuelle d'isolement d'enceinte	qui
	est équipée (du)
clapet de sécurité rapide.	

Figure 1. An example of splitting.

[1] une armoire de contrôle sensible à une élévation de température
[1'] une (armoire de contrôle) (sensible à une élévation de température)

The system then needs additional syntactical information, on the sub-categorization properties of adjectives. For example, it must have at its disposal the list of the adjectives likely to be built with the "*à*" preposition. Rather than a priori giving that list to the system, we have chosen to equip it with a procedure allowing it to build that list by itself, by the analysis of the corpus. That procedure is a very simple one. During a first pass, the procedure collects all the adjectives which appear in a predicative position followed by the "*à*" preposition. During a second pass, each time a splitting rule has just eliminated a sequence beginning with the "*à*" preposition, the system eliminates the eventual adjective which precedes it if owing to the so built list. The empirical analysis of that procedure shows how efficient it is. This is the most simple example of the Corpus-Based Endogenous Learning (CBEL)

procedures which have been integrated into LEXTER. More complex procedures have been implemented to acquire the following syntactical information:

- the list of nouns sub-categorizing the "*à*" preposition
- the list of nouns sub-categorizing the "*sur*" preposition
- etc. (the same with 12 different prepositions)
- the list of adjectives sub-categorizing the "*de*" preposition
- the list of adjectives sub-categorizing the "*à*" preposition
- the list of past participles sub-categorizing the "*de*" preposition

3.3 Interest and limits of CBEL

The perfecting of the principle of term locating by splitting makes appear, in some situations, the necessity of having at one's disposal syntactical information of sub-categorization. In order to give these information to the system, one approach could have consisted in building a lexicon gathering all this information for the whole lexical units concerned. In front of the extent and the difficulty of the collecting work that such a task implies, we turned ourselves towards an approach in which the system itself has to acquire these information from the corpus it analyzes. CBEL not only avoids any collecting phase a priori, for it draws its information from the corpus itself, but also allows the system to acquire some idiosyncratic syntactical characteristics of the language for special purpose. Let us notice that there is no cumulative effect. The system "forgets" all the information it has learnt at each new corpus analyzed.

We give an illustration of that remarkable adaptation property by an example. We return to the example of section 2.2. The CBEL procedure captures the idiosyncratic properties of association between adjectives and "*à*" complements which are made possible by the polysemy of the preposition. An exogenous method would only supply the "normal" properties of sub-categorization of adjectives. As an example, in one of our corpora, the "*disponible*" adjective is frequently used followed by a locative interpretation complement introduced by the "*à*" preposition (for example: "*Ce logiciel est disponible à la direction informatique*"). That adjective is not known as a sub-categorizer of that preposition. But the learning procedure collects that adjective, which will be rightly eliminated in cutting cases as the following one: "*les maquettes disponibles au département Etudes*". In that context, the system will not extract the "*maquettes disponibles*" group.

Of course the limits of CBEL are linked to its advantages. The price to pay for an automatic acquisition of syntactical information, without any intermediary human validation is double risked: the procedure can let relevant information pass through, it can also acquire false information. That is the reason why the perfecting of these procedures requires the adoption of experimental processes, with numerous tests on large-scale corpora, which ensure the global empirical validity of these procedures.

4. The Parsing module

At the parsing stage, LEXTER parses the maximal-length noun phrases isolated by the Splitting in order to grammatically break up each complex candidate term into a head and an expansion. According to a well-established principle of terminology (the principle of syntagmatic derivation), any complex term can be divided into two constituents: a constituent in head-position, representing more often a super ordinate concept (e.g. *analysis* in the term *syntactic analysis*), and a constituent in expansion-position, mentioning a specific attribute (e.g. *syntactic* in the term *syntactic analysis*).

The decomposition operation performed by the parsing module generates sub-groups, in addition to the MLNP, which are candidate terms, and allows to build a large terminological network (see the next module). The LEXTER parsing module is made up of parsing rules which indicate which sub-groups to extract from a MLNP, in head-position and in expansion-position, on the basis of grammatical sequence. One simple rule of the parsing module is given in figure 2 (H for head, E for expansion), with an instance of application.

Parsing rule [a]
noun$_1$ adj prep noun$_2$ -->**H** : noun$_1$ adj **H** : noun$_1$ **E** : adj **E** : noun$_2$
vanne manuelle d'isolement -->**H** : *vanne manuelle* **H** : *vanne* **E** : *manuelle* **E** : *isolement*

Figure 2. A simple parsing rule.

Some of the MLNP sequences are non-ambiguous: given such a sequence, it can be stated with a very high rate of certainty that only one parsing is valid. The corresponding parsing rules are called non-ambiguous rules. Parsing rule [a] is a non-ambiguous rule. Some of the MLNP sequences are ambiguous, that is, given such a sequence it cannot be stated with a sufficient rate of certainty that only one parsing is valid. Several binary decompositions compete, corresponding to several possibilities of prepositional phrase or adjective attachment. The disambiguation is performed by a corpus-based method which relies on endogenous learning procedures (Bourigault 1993). An example of such a procedure is given on figure 3.

Parsing rule [b]

noun$_1$ prep noun$_2$ adj

-->

Parse (1)	Parse (2)
H : noun$_1$	**H** : noun$_1$ prep noun$_2$
E : noun$_2$ adj	**H** : noun$_1$
H : noun$_2$	**E** : noun$_2$
E : adj	**E** : adj

Disambiguation procedure

To look in the corpus for non ambiguous occurrences of the sub-groups

(a) noun$_2$ adj

(b) noun$_1$ adj

(c) noun$_1$ prep noun$_2$

Then to choose :

if	the sub-group (a) has been found	choose Parse (1)
else if	the sub-groups (b) or (c) have been found	choose Parse (2)
else		choose Parse (1)

Figure 3. An ambiguous parsing rule with its disambiguation procedure

5. The Structuring module

The structuring module exploits the analysis provided by the Parsing module to organize all the candidate terms under a network format, known as "terminological network". This module links each analyzed complex candidate term to both of the candidate terms which constitute its head and its expansion. Thus the structuring module yields a very dense network of candidate terms that are connected to one another by

two types of oriented links: H-links and E-links. As an illustration, figure 4 gives an extract of a network of candidate terms which has been yielded by LEXTER from a power plant maintenance manual.

The building of the network is especially important within a terminology acquisition prospect, for it allows to underscore lists of terms that have the same term in common either in H-position or in E-position. One knows that such paradigmatic series are not rare at all in terminologies. Thus for each candidate term, the structuring module calculates a coefficient of relevance which is all the higher since the candidate term is more productive, or in other words, since it is a part of a greater number of candidate terms, either in H-position or in E-position. Taking the network of figure 4 as an example, "*vanne*" is a very productive candidate term; it is highly probable that this candidate term and the candidate terms which it is a part of, actually are terms of the subject-field.

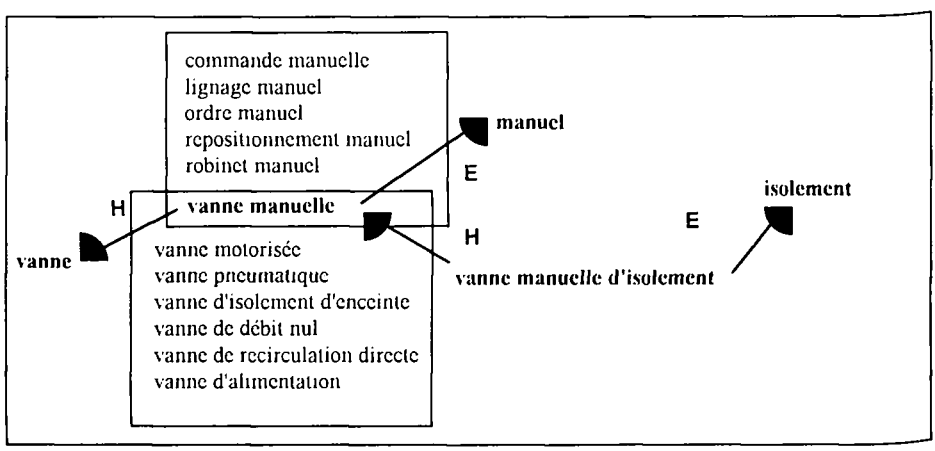

Figure 4. An example of a terminological network produced by LEXTER.

6. Applications

LEXTER was initially realized with the aim of answering the building and updating needs of the thesaurus exploited by an automatic text indexing system. At the Research and Development Division of the French Electricity Board, LEXTER is today mainly used in Electronic Document Management applications. We are especially working on the problem of the semi-automatic building of terminological index for electronic books and, more generally, of hypertext consultation system for large technical documentation (Gros and al. 1994). As we already

pointed it out, LEXTER is domain-independent and can be used on different kinds of technical documents. It is currently used on different corpora to build different kinds of terminological products.

We think that knowledge acquisition for knowledge based systems is also a favorable experimentation ground for such a terminology extraction software (Bourigault 1995). LEXTER has been used in a real application of knowledge acquisition, that is the SADE project. A brief description of this experimentation can be found in (Aussenac-Gilles and al. 1995). LEXTER will also be used in a project aiming at building a Terminological Knowledge Base (Meyer and al. 1992) on the field of nuclear power plant maintenance.

References

Aussenac-Gilles, N., D. Bourigault, A. Condamines and C. Gros 1995. "How can Knowledge Acquisition benefit from Terminology?", in: *Proceedings of the 9th Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'95)*. Banff.

Bourigault, D. 1992. "Surface grammatical analysis for the extraction of terminological noun phrases", in: *Proceedings of the 15th International Conference on Computational Linguistics (COLING '92)*. Nantes.

Bourigault, D. 1993. "An endogenous corpus-based method for structural noun phrase disambiguation", in: *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics (EACL'93)*. Utrecht.

Bourigault, D. 1995. "LEXTER, a Terminology Extraction Tool for Knowledge Acquisition from Texts", in: *Proceedings of the 9th Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'95)*. Banff.

Grefenstette, G. 1992. "Use of syntactic context to produce term association lists for text retrieval", in: *Proceedings of the 15th International Conference on Research and Development in Information Retrieval (SIGIR'92)*. Copenhagen.

Gros, C., D. Bourigault and J. L. Vuldy 1994. "Linguistic-Based toolbox for Hypertext Automatic linking on Large Technical Documentation", in: *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*. Gaitherburg.

Meyer, I., D. Skuce, L. Bowker and K. Eck 1992 "Toward a new generation of terminological resources: an experiment in building a terminological knowledge base", in: *Proceedings of the 14th International Conference on Computational Linguistics*. Nantes.