Compiling a "classical" explanatory combinatorial lexicographic description into a relational database

Jacques Steinlin†, Sylvain Kahane†, Alain Polguère‡
† LaTTice – Université de Paris 7, ‡ OLST – Université de Montréal
jsteinli@linguist.jussieu.fr, sk@ccr.jussieu.fr, alain.polguere@umontreal.ca

Abstract

This paper deals with the processing of a formal electronic dictionary of French centered around the description of semantic derivations and collocations. The purpose of our project was to obtain a more flexible format for this dictionnary, the DiCo, which has a traditional structure. By compiling the DiCo, we get an object model which, in turn, can be stored in an SQL database. We show how the object model allows us to more easily access the data and gives us a new perspective on the structure of dictionary. This latter can be completly reorganized following new axes like lexical function links.

1 Introduction

The present work is part of a lexicographic project that targets an electronic dictionary of French centered on semantic derivations and collocations. The dictionary is developed within the framework of the Explanatory Combinatorial Lexicology, the lexical module of the Meaning-Text Theory ((Mel'čuk, 1974), (Žolkovskij & Mel'čuk, 1965), (Mel'čuk *et al.*, 1995)) and the project is supervised by Igor Mel'čuk and Alain Polguère at the OLST–Université de Montréal. The electronic dictionary, the DiCo, is designed to serve as a generic lexicographic description from which one can produce i) "computable" databases such as the one presented here and ii) "general public" descriptions such as the *Lexique Actif du Français* or *LAF* (Polguère, 2000). The DiCo possesses a rather traditional structure, inspired by paper dictionaries, and adopted by numerous electronic dictionaries (TLFi (Dendien & Pierrel, 2003), eEH (Arregi *et al.*, 2003)). The entries of the dictionary are vocables (including idioms). Every vocable is a set of lexical units corresponding to its different senses. Our goal was to propose for the DiCo a richer and more flexible architecture to improve the access to the data. On one hand, this architecture aims at improving the query possibilities of the DiCo for human users and at making the lexicographic information exploitable by natural language processing systems (especially in natural language generation). On the other hand, it should facilitate all the tasks of updating and development of the dictionary, that is the lexicographic tasks. For that reason, we chose to parse the existing dictionary, the DiCo, and then to feed a relational database via an object representation. In this paper, we present the result of this modeling, the DiCobjet, and show the advantages of such a structure from a theoretical point of view as well as from a practical one. This structuring can be compared with the one which is used in WordNet (Fellbaum, 1998). However, as we shall see, the network resulting from the analysis connects not only lexical items by means of lexical function relations, but also encodes all lexicographic information that is associated with each lexical unit (such as, for example, the subcategorization frame). We first introduce the DiCo as it is developed by lexicographers, then we describe its compiled structure and its representation. We end with a discussion of the theoretical and practical advantages obtained by our modelization.

2 Overview of the DiCo

As we said, the DiCo adopts a structure close to the structure of standard paper dictionaries, that is, a structuring made up of lexicographic entries. This structuring is required by the lexicographer, who wants to have a global vision of the bulk of information related to each lexical unit. The description inside an entry consists of nine main fields, each corresponding to a specific type of information, as we can see on Figure 1.

Field	Value
word	ENGUEULADE
lexicographic number	1
grammatical properties	nom, fém, "fam"
semantic label	communication langagière
semantic formula	~ DE L'individu X [VISANT L'individu Y] POUR LE fait Z
subcategorization frame table	X = I = de N, A-poss Y = II = -- Z = III = Prep-pour N | Prep-pour = {_à propos de_, _au sujet de_, pour}, pour V-inf-passé
synonymy	{QSyn} "fam" savon ; "soutenu" admonestation, remontrance, réprimande ; blâme
lexical functions	{QAnti} compliment, félicitation ; flâtterie {V0} engueuler {Magn} belle, bonne, sacrée | antepos < majeure | postpos {Oper213} essuyer [ART ~ _de la part de_ N=X Prep-pour N=Z] {Oper23} subir [ART ~ Prep-pour N=Z]
Phraselogy	

FIG. 1 – Partial description of the lexical unit ENGUEULADE$_1$ (='argument', 'bawling out')

The first two fields contain the name of the vocable and the lexicographic number of the lexical unit. The **grammatical properties** field records the part of speech of the lexical unit as well as other information relative to its cooccurrence : stylistic label (ex : "fam"), graphic variants, inflection constraints (ex : "no plural"), and so on. The **semantic label** field position the lexical unit in a semantic hierarchy used to account for the central semantic behavior of lexical units (for a detailed description of the DiCo semantic labeling, see (Polguère, 2003)). The **semantic formula** field enumerates the semantic actants of the lexical item and possibly supplies them with a semantic label. The **subcategorization frame table** describe how semantic actants are to be expressed in structures that are syntactically controled by the lexical unit. **Synonymy** and **Lexical Functions** fields describe lexical function links controled by the lexical unit. A lexical function's name is written between braces and is followed by a value list. Lexical functions like **Magn** or **Oper**$_i$ allow to describe the collocations in which the lexical unit occurs : *une belle engueulade*, *une sacrée engueulade* or *subir une engueulade*. Finally the **phraseology** field lists all the full idioms that formally contain the lexical unit. It is empty here, ENGUEULADE$_1$ having no associated full idioms. In practice, a relational mono-table database is used for the storage of the DiCo. Every entry is a record, each row corresponding to a field inside the record. Beyond this first level of structuring, we find a second level of structuring, which is syntactic. Every field is written according to some syntactic and typographic conventions. Thus, it becomes possible to identify automatically the keys of the description. We were able to design a compiler to translate automatically the DiCo into the DiCobjet. A similar work was made within the framework of the project Papillon for the translation of the DiCo in a XML format ((Lapalme & Sérasset, 2003)).

3 Object Modeling of the DiCo

3.1 General overview of the object modeling

As we can see, the fields described in the previous section contain lists of information items which we can split into fields and so on. This is the compiler purpose. The content of the DiCo is fed into the compiler as a tab-separated text. The compiler parses the file and produces a semantic representation of the DiCo, the DiCobjet[1]. One of the objectives that we set ourselves in compiling the DiCo was to obtain the most detailed representation as possible. But in fact, the modeling only consists in bringing to light (by clarifying it) the structure of the DiCo. So, by examining again figure 1, we notice that the lexical function field contains a list of lexical functions. A lexical function is supplied with a value list linked to the lexical function formula (for example, **Oper**$_{213}$). Each of these values (a support verb in this case) can be organized in turn in fields : the value itself, its style label (`"soutenu"`), its subcategorization frame, a constraint (for example `postpos` for an adjective), and so on.

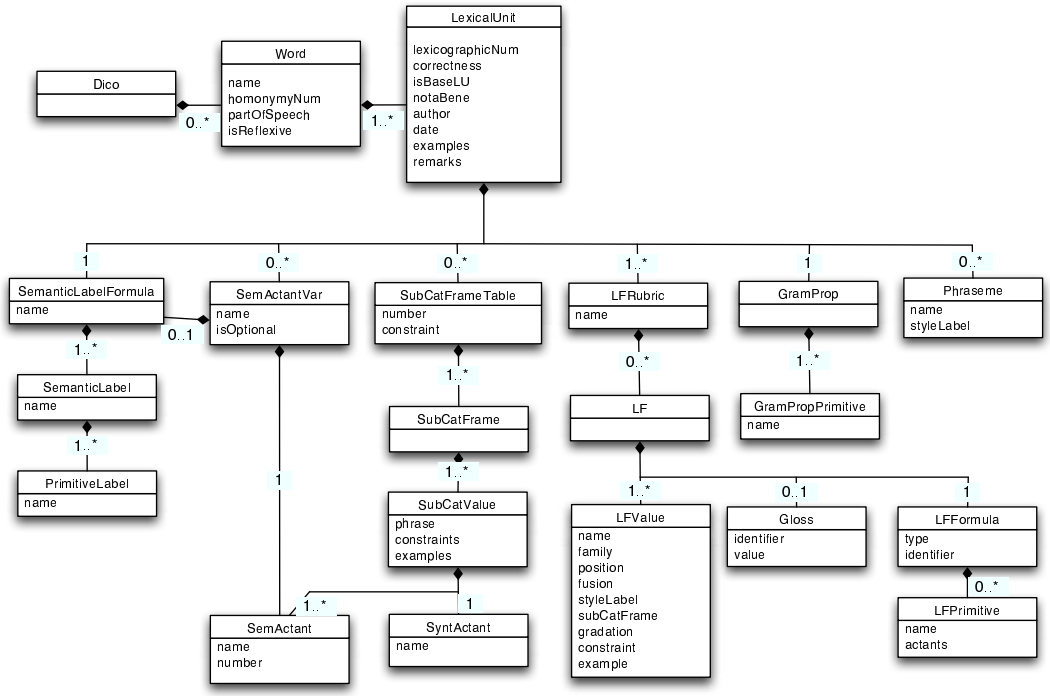

FIG. 2 – DiCo Object Diagram

The work of the compiler thus consists in identifying the atoms of information and organizing them in an adequate data structure. The data structure representing the DiCo is the DiCobjet (see the diagram, figure 2). We represented the classes (that is, the definition of objects) by boxes. The upper part of the box is reserved for the name of the class. The lower part enumerates the attributes which are of primitive type (integer, boolean, character strings). Other attributes, which have classes for value, are represented by the relations between the classes. These relations, although directed, are bidirectional. We call them aggregate relations. Such a relation means that the class being on the side of the diamond owns one or several classes as attributes. On the other side of the relation, we wrote the number of objects expected in the aggregation. Figure 2 must be read in the following way : a DiCobjet is a set (of size superior or equal to 0) of Word objects. The Word class is characterized by attributes (only the attribute "name" being compulsory) and a set of LexicalUnit objects (the set must include at least one element). A LexicalUnit

[1] *Semantic representation* is a computer science concept. From the computer point of view, the input is just a character string. In compiling it, the character string is translated into objects which have a semantic value. For example, the input string {Magn} becomes a LFFormula object (and therefore it has some specific properties).

is described in turn as a container of collections of complex objects : SemanticLabelFormula, a set of semantic actants (SemActantVar), a set of tables of subcategorization frame (SubCatFrameTable), a set of Phrasemes, one grammatical property (GramProp, which is a set of grammatical properties primes) and LFRubric. LFRubric, besides being defined by a name, includes a set of objects of type LF, this last class consisting of the grouping of LFFormulas (what we usually call a Lexical Function), of a Gloss (the paraphrase in natural language of the lexical function) and of a set of LFValue (the values returned by the lexical function applied to the described lexical unit). Finally, LFFormula is a tree structure combining some LFPrimitive. The class LFValue contains many attributes. Among these, the value strictly speaking (for example, *flatterie*) is a string. This modeling is not satisfactory as far as the value is in fact a more complex object and might be treated as a LexicalUnit. In the case of a complex lexical item like an expression, we give its syntactic composition with a dependency tree. We have to emphasize that the aggregate relation does not foresee the orientation of the link between objects. If SubCatFrameTable is clearly an attribute of a LexicalUnit, it is also possible to go back up from a given SubCatFrameTable to the LexicalItem objects which possess it. In the conclusion we will return to the repercussions of this property of the model on the conception we have of the dictionary.

3.2 Exploitation

The advantages of object modeling are numerous. It becomes possible to do sophisticated queries and to filter the results of these to keep only the relevant information (instead of the complete records). For example, we can retrieve the list of the lexical units having at least two actants whose first one is realized by the surface forms `de N, A-poss` or the lexical items having at once two semantic actants and one `Real@`. To allow this enhancement of the query power, it is necessary to add the possibility of making queries under other viewpoints that the one imposed by the structuring in words. We can consider, for example, the dictionary from the lexical functions point of view. The current version of the DiCo gives 143 different values for the lexical function $\mathbf{Oper}_1$ (Figure 3 gives a sample). We get back also 48 different lexical functions returning the value *donner* (see figure 4).

nb	Valeur	nb	Valeur	nb	Valeur	nb	Valeur
64	avoir	14	pousser	1	aller	1	être pris
42	être	11	émettre	1	battre	1	jauger
27	faire	10	constituer	1	célébrer	1	occuper
20	posséder	9	_faire preuve_	1	comprendre	1	partir
17	ressentir	8	se trouver	1	couvrir	1	peupler
15	éprouver	(...)	(...)	1	être (présent)	1	proférer

FIG. 3 – $\mathbf{Oper}_1$ values in decreasing frequency

nb	FL	nb	FL	nb	FL
10	{CausFunc1}	3	{Liqu1Oper1}	2	{Labreal21}
7	{Oper12}	3	{Son}	2	{Liqu1Real1}
6	{Oper1}	2	{Caus1Func0}	2	{Real31–I}
5	{Oper2}	2	{Caus de nouveau Func1}	2	{Real31–II}
4	{Oper13}	2	{Caus1Func0}	2	{Realo–II}
4	{Real12}	2	{Fact1}	1	{PredAble2}
3	{Caus3Func0}	1	{Fact21-production}	...	...

FIG. 4 – Lexical Functions returning the lexical unit *donner* in the current version of the DiCo

So, as it was requested by ((Grossmann & Tutin, 2003)), we can now try to discover regularities among the values since we can now group them by the semantic label of the lexical unit they are linked to. For

example, we can see that many lexical units labeled with the semantic label *sentiment* (*feeling*) have the lexical function **IncepPredPlus** returning the value *augmenter* (*to increase*). Similarly, we can compare values returned by the LF's $\mathbf{Oper}_1$ and $\mathbf{Real}_1$ or $\mathbf{Fact}_0$ and $\mathbf{Real}_1$. It is also possible to check wether a lexical unit is, at the same time, an argument of many LF's such as $\mathbf{Oper}_1$ and $\mathbf{Real}_1$[2].

Further applications could possibly be the use of the DiCobjet API to program procedures of planification for new entries based on related entries. Not only the trivial **QSyn** relation could be used, but also lexical units sharing the same semantic label.

3.3 Practical and theoretical contribution of object modeling

The DiCobjet was automatically extracted from the DiCo. This was possible thanks to the usage of strict conventions in the editing of the DiCo (only some minor adjustments were necessary to reduce the existing ambiguities). Notice that one of the main requirements of the lexicographers is to be able to continue to develop the DiCo with its initial format (or an equivalent one). Indeed, as we said earlier, the compilation of a dictionary cannot amount to a blind filling of predefined fields and the lexicographer must be able to read a dictionary entry taken as a whole. As is expected, the object modeling of the dictionary increases the access to the data of the DiCo, by allowing varied queries. More crucially the result returned by these queries is not any more inevitably the entry, it can be a value of lexical function or a set of subcategorization frames. Besides the query and the exploitation, the modeling facilitates the revision of the DiCo by enabling an easy check of the homogeneity of every type of information. It also facilitates its development by the processing on the data and by allowing for example to combine the information of several entries to build the skeleton of a new entry. The object modeling of the dictionary increases not only the possibilities of data mining. The most important and least expected result from the project is that our vision of the dictionary changed. For example, at the surface level, the dictionary is no longer a flat collection of entries. The object model can be taken in fact by any end and the DiCo can thus be completly turned upside down. We can get into the DiCo by the FL object and produce for every lexical function an entry giving the lists of the values for every keyword. We have then a much more semantic vision of the dictionary, where we see how senses like the intensification, the causation, the realization, etc. are realized in very different ways. We do not have to deal any more with a dictionary of 2000 entries as in the beginning, but with a dictionary of several dozens thousand entries. The online dictionary can be reached at the following address : `http ://www.olst.umontreal.ca/dicouebe/`.

Références

ARREGI X., ARRIOLA J., ARTOLA X., DIAZ DE ILARRAZA A., GARCÍA E., V. L., SARASOLA K., SOROA A. & URIA L. (2003). Semiautomatic conversion of the euskal hitzegia basque dictionary to a queryable electronic form. *Traitement Automatique des langues, TAL, 44 n˚2, 107-124.*

DENDIEN J. & PIERREL J. M. (2003). Le trésor de la langue française informatisé. un exemple d'informatisation d'un dictionnaire de langue de référence. *Traitement Automatique des langues, TAL, 44 n˚2, 11-37.*

C. FELLBAUM, Ed. (1998). *WordNet, An Electronic Lexical Database*. MIT Press.

GROSSMANN F. & TUTIN A. (2003). Quelques pistes pour le traitement des collocations. In F. GROSSMANN & A. TUTIN, Eds., *Les collocations, analyse et traitements*. Revue Française de linguistique appliquée.

LAPALME G. & SÉRASSET G. (2003). Batch creation of Papillon entries from DiCo. In *Workshop Papillon*.

MEL'ČUK I. (1974). Opyt teorii lingvisticheskih modelej "smysl ↔ tekst". semantika, sintaksis. *Nauka*.

[2] In the DiCo, 229 lexical units have $\mathbf{Oper}_1$ values while 99 have $\mathbf{Real}_1$ values ; but only 22 have both.

MEL'ČUK I., CLAS A. & POLGUÈRE A. (1995). *Introduction à la lexicologie explicative et combinatoire*. Duculot.

POLGUÈRE A. (2000). Towards a theoretically-motivated general public dictionary of semantic derivations and collocations for french. In *Proceedings of EURALEX 2000*.

POLGUÈRE A. (2003). Étiquetage sémantique des lexies du DiCo. *Traitement Automatique des langues, TAL, 44 n˚2*.

ŽOLKOVSKIJ A. & MEL'ČUK I. (1965). O vozmožnom metode i instrumentax semantičeskogo sinteza. *Naučno-texničeskaja informacija*, **6**, 23–28.