Dictionaries for language processing. Readability and organization of information

Eric Laporte

DICTIONARIES FOR LANGUAGE PROCESSING
READABILITY AND ORGANIZATION OF INFORMATION

Éric Laporte
Universidade Federal do Espírito Santo
Université Paris-Est
eric.laporte@univ-paris-est.fr

Abstract: What makes a dictionary exploitable in Natural Language Processing (NLP)? We examine two requirements: readability of information and general architecture, and we focus on the human tasks involving NLP dictionaries: construction, update, check, correction. We exemplify our points with real cases from projects of morpho-syntactic or syntactic-semantic dictionaries.
Keywords: NLP dictionary. Readability of information. Dictionary management. Morpho-syntactic dictionary. Syntactic-semantic dictionary.

What makes a dictionary exploitable in Natural Language Processing (NLP)? In this paper, we examine two requirements: readability of information and general architecture, and we focus on the human tasks involving NLP dictionaries: construction, update, check, correction. We exemplify our points with real cases from projects of morpho-syntactic or syntactic-semantic dictionaries.

Section 1 is an introduction to NLP dictionaries. In Section 2, we study the readability of syntactic data provided by dictionaries. In Sections 3 and 4, we compare general architectures and argue in favour of grouping all data related with a given lexical entry, or to a homogeneous set of lexical entries. The conclusion brings final considerations.

1. What are NLP dictionaries?

In this article, what we call an NLP dictionary is a linguistic data set that lists words and provides information about them in such a way that exploitation in NLP applications is possible.

We will restrict our focus on those NLP dictionaries that include in their content either syntactic-semantic information, e.g. about complements of verbs, or information related to morpho-syntax and inflected forms. Syntactic-semantic information is required to recognize the structure of input sentences, and inflectional information is necessary for

identifying inflected forms of words. In practice, this restriction excludes WordNets (MILLER, 1990; FELLBAUM, 1998) and most ontologies, since both provide parts of speech of words and semantic relations between them, but almost no syntactic or inflectional features.

How are dictionaries used in NLP? The most obvious use is the final, computational one: some systems of translation or information extraction, for instance, look up dictionaries while they are operating, and determine their output depending on what they retrieve. At the other end, the beginning of the processing chain involves linguists that build, check, correct and update the dictionaries. The quality of the systems depends on these human tasks. In fact, the human tasks on the beginning of the chain and the computational tasks on the final end are often performed on distinct versions of the dictionaries, with distinct file formats, as shown in Fig. 1. Formats for human use are typically more compact and readable, while those for computational use are more voluminous, may distribute data in several files, and may be totally unintelligible for humans. Dictionaries in the first type of format are automatically converted into the second type: this operation is symbolized by the solid arrow in Fig. 1 and called ‘compilation’. In the reverse direction, dotted arrows show how observation of systems’ output can provide feedback on dictionaries and guide human tasks so as to enhance performance.

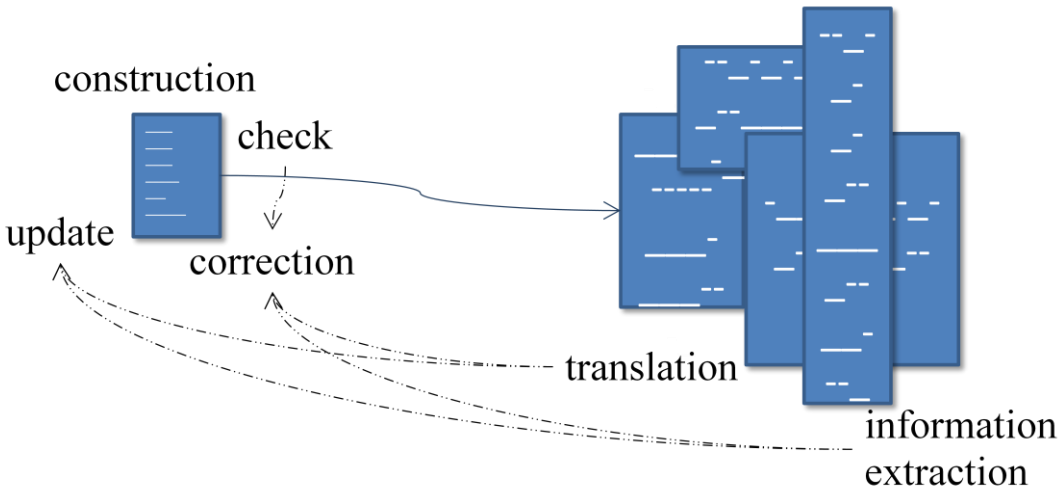

Fig. 1. Uses of NLP dictionaries

Another general feature of NLP dictionaries is that they look completely different from dictionaries intended for human readers (cf. Fig. 1-9). In NLP dictionaries, information must be encoded in a computer-compatible form, and cannot appear in the form of text: for instance, definitions and examples in the form of text are not directly exploitable by current systems. By contrast, in dictionaries for human readers, most information takes the form of textual definitions and examples.

The human tasks of construction, check, correction and update of NLP dictionaries present practical difficulties which makes them notoriously costly and time-consuming:

a) lexical entries are numerous, in the order of magnitude of 10 000 verbs, 10 000 adjectives and 50 000 nouns;
b) automatic aids are little efficient;
c) words show an impressive diversity of behaviour.

Therefore, a major requirement for exploitability of NLP dictionaries is that their format and organization should facilitate human management tasks. In the rest of this article, we study how such tasks are simplified when

a) syntactic constructions are encoded in a readable way
b) and dictionaries have an appropriate general architecture.

We will take our examples from two kinds of sources:

a) projects of large-coverage NLP dictionaries, such as, for English, FrameNet (FILLMORE, ATKINS, 1994), VerbNet (KIPPER *et al.*, 2006), Comlex (GRISHMAN *et al.*, 1994), and for French, the Lexicon-Grammar (GROSS, 1975, 1984), Le*fff* (SAGOT *et al.*,2006), Dicovalence (VAN DEN EYNDE, MERTENS, 2003);
b) international recommendations and *de facto* standards: EAGLES (CALZOLARI *et al.*, 1996), LMF (FRANCOPOULO *et al.*, 2006), Dela (COURTOIS, 1990).

2. Encoding syntactic constructions

Syntactic dictionaries formalize syntactic constructions in quite different ways. In FrameNet (Fig. 2), each argument is represented in a cell. For example, the PP[by] cell in the first row stands for *by the same families* in:

(1) *Many shops have been managed by the same families for years*

Duration	Operator	System	
PP[for] Dep	PP[by] Dep	NP Ext	
Manner	Operator	System	
AVP Dep	NP Ext	NP Obj	
Manner	Operator	System	Time
AVP Dep	NP Ext	NP Obj	AVP Dep

Fig. 2. Syntactic constructions of *manage* in FrameNet

In the Lexicon-Grammar (Fig. 3), features are labeled by short formulae. For example, the *N0 V N1 en N2* feature stands for the construction of sentences as *Le Conseil rassemble des experts en un comité* “The Council gathers experts into a committee”. VerbNet and Dicovalence have similar conventions.

+	N1 =: N-hum
-	N1 être V-n
+	N0 V N2
+	N0 V N1 Loc N2 source Loc N3 destination
+	N0 V N1 en N2
-	N0 V N1 de N2 source
-	N0 V N1 Loc N2 source, Loc # de
+	Loc N3 =: dans N3 destination
-	Loc N3 =: sur N3 destination
+	Loc N3 =: contre N3 destination
-	Loc N3 =: à N3 destination

Fig. 3. Syntactic constructions of *rassembler* “gather” in the Lexicon-Grammar

In Comlex (Fig. 4), each syntactic construction is represented by a name and two parenthesized formulae. For example, the *part-wh-s* construction is that of:

(2) *I found out how I can meet Kevin*

part-that-s
*:cs ((part 2 :adval (" ")) (s 3 :that-comp required))
:gs (:subject 1, :part 2, :comp 3)
part-wh-s
*:cs ((part 2 :adval (" "))(s 3 :q (wheth how)))
:gs (:subject 1, :part 2, :comp 3)
part-what-s
*:cs ((part 2 :adval (" "))(s 3 :q (what 4) :omission 4))
:gs (:subject 1, :part 2, :comp 3)

Fig. 4. Syntactic constructions of *find out* in Comlex

These three styles of encoding syntactic formulae are not equally readable. Two factors of readability are relevant:

a) how the syntactic construction is linearized,
b) how much formulae reconcile compactness and mnemonicity.

2.1. Linearization

By 'linearization' we mean the encoding of a complex object (a syntactic construction) into a sequential string of symbols (a formula).

The syntactic formula is more readable when it contains symbols for the elements that are mandatory in the syntactic construction. This is the case for the *N0 V N1 en N2* feature in Fig. 3. By contrast, the verb is not represented in *PP[for] PP[by] NP* (Fig. 2) or in the *part-wh-s* formula of Fig. 4, though it is mandatorily present in the constructions, as can be seen by comparing (1) and (2) with:[1]

[1] The asterisk * marks sequences as being unacceptable as sentences.

**Many shops by the same families for years*

**I how I can meet Kevin*

The absence of the verb in these formulae stems probably from a concern with parsimony, since the main verb is mandatory in most sentential constructions. Indeed, the only reason to include a symbol for the verb in syntactic formulae is readability; but this reason is strong enough.

Another factor of readability is when the symbols in the formula are presented in a sequential order that matches an acceptable ordering in the syntactic construction. Again, this is the case for the *N0 V N1 en N2* feature in Fig. 3: the sequential order is the same as in the sentence *Le Conseil rassemble des experts en un comité*. The same holds for the *part-wh-s* formula of Fig. 4: in **:cs ((part 2 :adval (" "))(s 3 :q (wheth how)))*, the subformulae for the particle *(part 2 :adval (" "))* and for the complement *(s 3 :q (wheth how))* appear in this order; and so do, in *:gs (:subject 1, :part 2, :comp 3)*, the subject, particle and complement. Meanwhile, the order of the arguments in *PP[for] PP[by] NP* (Fig. 2) is not accepted in sentences:

**For years have been managed by the same families many shops*

2.2. Compactness

The display format of Fig. 5 allows for cross-tabulating on a single screen dozens of lexical entries with dozens of features. Thus, dictionary authors encode an entry with a description of comparable ones in front of them, provided entries are grouped into sufficiently homogeneous classes. This format facilitates encoding. It requires that each syntactic feature can be displayed on the screen, and therefore should be encoded in the form of a brief label, up to, say, 30 characters, as in the Lexicon-Grammar, FrameNet and VerbNet. In the Comlex format, the names for syntactic constructions, e.g. *part-wh-s* in Fig. 4, meet this requirement of compactness, but they convey almost no explicit information, whereas complete formulae typically occupy 2 to 4 lines. In the LMF format, an ISO standard, the representation of a syntactic construction is even more verbose and often spans over 30 lines (LAPORTE *et al.*, 2013).

N0 =: Nhum	N0 =: Nnr	N0 =: V-inf W	N0 être V-n	Ppv	Ppv =: se figé	Ppv =: en figé	Ppv =: y figé	Ppv =: Neg	**<ENT>**	Nég obl	Aux =: avoir	Aux =: être	N0 être V-ant	N0 être Vpp	N0 V de N0pc	[extrap]	N actif V N0	<OPT>
+	-	-	-	<E>	-	-	-	-	barboter	-	+	-	-	-	-	+	~	Max barbote dans l'eau
~	~	~	~	<E>	-	-	-	-	basculer	-	+	-	~	~	~	~	~	la chaise bascule
-	-	-	-	<E>	-	-	-	-	battre	-	+	-	-	-	+	-	~	Son coeur bat
-	-	-	-	<E>	-	-	-	-	béer	-	+	-	+	-	-	+	~	Sa bouche bée
-	-	-	-	<E>	-	-	-	-	blouser	-	+	-	+	-	-	-	~	Le chemisier blouse
~	~	~	~	<E>	-	-	-	-	boiter	-	+	-	~	~	~	~	~	Cette chaise boite
~	~	~	~	<E>	-	-	-	-	bomber	-	+	-	~	~	~	~	~	La voiture bombe
~	~	~	~	<E>	-	-	-	-	boucler	-	+	-	~	~	~	~	~	Le programme boucle
-	-	-	-	<E>	-	-	-	-	bouffer	-	+	-	+	-	-	-	~	Ses manches bouffent
~	~	~	~	<E>	-	-	-	-	bouger	-	+	-	~	~	~	~	~	La dent bouge
~	~	~	~	<E>	-	-	-	-	bouillir	-	+	-	~	~	~	~	~	L'eau bout à cent degrés

Fig. 5. Sample of a Lexicon-Grammar table (BOONS *et al.*, 1976).

One of the methods implemented to reduce the size of syntactic formulae without losing information is to specify values and not attributes. Most information provided in syntactic formulae is naturally expressed in the form of attribute-value pairs like [part of speech/noun], [preposition/*by*] or [tense-mood/past participle]. Once the value (noun, *by* or past participle) is explicitly specified, the corresponding attribute (part of speech, support verb or tense-mood) is much less informative and can often be omitted. For example, *PP[for] PP[by] NP* and *N0 V N1 en N2* specify the values *for, by* and *en* “into”, but do not explicitly state that they are prepositions.

The LMF standard does not take advantage of this trick, since it is expressed in the XML language, in which both elements of an attribute-value pair are mandatory, as in:

```
<feat att="partOfSpeech" val="noun"/>
```

The size of syntactic formulae can also be limited by avoiding multiple levels of parentheses. In addition to making formulae verbose, piled parentheses are a significant obstacle to reading for humans, as it happens with feature-structures. Take for example the *LGLex* dictionary (CONSTANT, TOLONE, 2010), automatically generated from the Lexicon-Grammar for use in syntactic parsers: it uses parentheses to show the structure of syntactic constructions (Fig. 6), which is convenient for computer programs, but less readable than Fig. 5. The Lexicon-Grammar dispenses with almost

all parentheses. For instance, if we specify syntagm boundaries in *N0 V N1 en N2*, we obtain something like *(N0 (V N1 (en N2)))*: we have added only obvious information, and the result is slightly less readable.

```
args=(const=[pos="0",dist=(comp=[cat="NP",introd-
prep=(),nothum="true",origin=(orig="N0 =: Nnr"),introd-
loc=()],comp=[cat="NP",hum="true",introd-prep=(),origin=(orig="N0 =:
Nnr"),introd-loc=()],comp=[cat="comp",introd-prep=(),origin=(orig="N0 =:
Nnr",orig="N0 =: Qu P"),mood="ind",introd-
loc=()],comp=[cat="comp",introd-prep=(),origin=(orig="N0 =:
Nnr",orig="N0 =: Qu P"),mood="subj",introd-
loc=()],comp=[cat="inf",introd-prep=(),origin=(orig="N0 =: Nnr"),introd-
loc=()],comp=[cat="leFaitComp",introd-prep=(),origin=(orig="N0 =: le fait
Qu P"),introd-loc=()],comp=[cat="leFaitInf",introd-
prep=(),origin=(orig="N0 =: V1-inf W"),introd-
loc=()])],const=[dist=(comp=[cat="NP",introd-
prep=(),nothum="true",origin=(orig="N1 =: N-hum"),introd-
loc=()],comp=[cat="NP",hum="true",introd-prep=(),origin=(orig="N1 =:
Nhum"),introd-loc=()]),pos="1"])
```

Fig. 6. Part of an entry of *LGLex*.

Summing up our observations about readability of syntactic formulae, most styles of encoding disregard basic techniques that would tend to increase readability. Not even international standards take into account this practical requirement which is crucial for authors of NLP dictionaries.

3. Architectures of syntactic dictionaries

3.1. Database table architecture

A mere glance at a Lexicon-Grammar table (Fig. 5) shows that it has the same architecture as a database table. Lexical entries are visually easy to identify and to compare with one another: they are the rows of the table. Similarly, syntactic-semantic

features are represented by the vertical alignment of their values, i.e. the columns. This tabular format directly relates entries with features.

A variant of this format is a table (Fig. 7) that relates classes of entries with features. When the feature is shared by all the members of the class, the common value is displayed, e.g. as "+" and "-" signs in Fig. 7, where all features are binary. The special symbols "o" and "O" mean that the members of the class do not share the same value for the feature, which must then be encoded in a table like that of Fig. 5.

	V_37M5	V_37M6	V_38L	V_38L0	V_38L1	V_38LD	V_38LH	V_38LHD	V_38LHR	V_38LHS	V_38LR	V_38LS	V_38PL	V_38R	V_38RR	V_39
N0 V N1 en N2	?	?	?	?	?	?	?	?	?	?	?	?	+	?	?	?
N0 V N1 entre N2pl obl	-	-	-	-	-	-	-	-	-	-	-	-	o	-	-	-
N0 V N1 et N2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
N0 V N1 Loc N1pc W	?	?	?	?	O	?	?	?	?	?	?	?	o	o	?	?
N0 V N1 Loc N2	?	?	?	o	?	-	-	-	+	-	+	?	?	?	?	-
N0 V N1 Loc N2 destination	?	?	-	-	?	+	?	+	?	?	?	?	?	?	?	?
N0 V N1 Loc N2 source	?	?	-	?	?	?	o	?	?	+	?	+	?	?	?	?
N0 V N1 Loc N2 source Loc N3 destination	?	?	+	?	?	?	+	-	-	-	?	o	?	?	?	?
N0 V N1 Loc N2 source, Loc # de	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
N0 V N1 Loc N2 V1-inf W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Fig. 7. Sample of a table of verb classes (TOLONE, 2012).

The general architecture of FrameNet and of the Lexicon-Grammar involves tables of lexical entries and tables of classes. To insert a new entry into these dictionaries, one can follow two steps:

a. determine the class where the entry fits, and insert it there (this automatically sets for the entry the features shared by all the class);
b. set the other features of the entry.

All data are directly connected to a lexical entry or to a class.

3.2. Architecture with syntactic constructions

In the LMF standard and in the Le*fff* dictionary, the central notion is that of syntactic construction. All features are linked to syntactic constructions. The insertion of an entry into an LMF dictionary involves the following steps:

a) compare the syntactic constructions of the entry with already encoded constructions;
b) if exactly the same construction is found, link it to the entry;
c) if it is not found, create it, encode its features according to the entry, and link the construction to the entry.

This framework and this style of encoding pose two maintenance problems. Firstly, the encoding of a lexical entry is not independent from others, since step one implies browsing other entries. Secondly, information about arguments is systematically duplicated. Take for example two constructions of the Portuguese verb *impedir* "prevent", exemplified by the following sentences:

O barulho impede o sono "Noise prevents sleep"
O barulho impede Theo de dormir "The noise prevents Theo from sleeping"

Distributional features can only occur at the level of syntactic constructions. Therefore, the distribution of the subject (noun phrases denoting human or non-human entities, and sentential subjects), which is the same for both constructions, as is usually the case, must be repeated. This introduces redundancy in the dictionary. The same holds for other argument-specific features (LAPORTE *et al*., 2013).

Again, the architecture adopted by an international standard is unhandy for dictionary authors.

4. Architectures of inflectional dictionaries

Let us turn to dictionaries providing information related to morpho-syntax and inflected forms. Such information is necessary for identifying inflected forms of words in texts. There is a traditional distinction between lemma dictionaries (Fig. 8) and full-form

dictionaries (Fig. 9). Lemma dictionaries contain only lemmas and are used to automatically generate inflected forms according to the codes provided in entries.

púlpito,N001
pulsação,N102
pulsante,A301
pulsão,N102
pulsar,N004
pulsar,V005

Fig. 8. Sample of Dela-PB (MUNIZ *et al.*, 2005)

púlpito,púlpito.N:ms
púlpitos,púlpito.N:mp
pulquérrima,pulcro.A:Sfs
pulquérrimas,pulcro.A:Sfp
pulquérrimo,pulcro.A:Sms
pulquérrimos,pulcro.A:Smp
pulsa,pulsar.V:P3s
pulsa,pulsar.V:Y2s
pulsação,pulsação.N:fs
pulsações,pulsação.N:fp
pulsada,pulsar.V:K
pulsadas,pulsar.V:K
pulsado,pulsar.V:K
pulsados,pulsar.V:K
pulsai,pulsar.V:Y2p
pulsais,pulsar.V:P2p
pulsam,pulsar.V:P3p
pulsamos,pulsar.V:J1p

Fig. 9. Sample of Delaf-PB (MUNIZ *et al.*, 2005)

4.1. Lemma-based architecture

In the lemma-based architecture (Fig. 8), the central notions are those of lemma and lexical entry. Codes are assigned to lemmas. The Dela format (COURTOIS, 1990) recommends this architecture as appropriate to manual maintenance, because the encoding of each entry is independent from others, and the notion of lexical entry is essential to linguistic analysis.

4.2. Full-form architecture

The full-form architecture (Fig. 9, without the lemmas) is organized around the notion of inflected form: codes are assigned to inflected forms. It is the only architecture of

morpho-syntactic dictionaries mentioned in the EAGLES Guidelines (CALZOLARI *et al.*, 1996), which recommend best practices and propose standards for computational lexicons and other subjects. In the samples of dictionaries provided in the Guidelines, the lemma of the inflected forms is not even included.

In inflectional languages, dictionaries with this architecture are unavoidably redundant, as it is obviously shown by the repetition of the verb stem on Fig. 9. Such redundancy causes problems of manual maintenance: for example, in Portuguese, the insertion or update of a single verb would involve the edition of about 70 forms.

4.3. Architecture based on lemmas and rules

Information related to morpho-syntax and inflected forms can be organized according to an alternative solution: a lemma-based part generates underlying representations of inflected forms; rules produce surface forms. Each rule is *a priori* applicable to any form, in opposition to the lemma-based architecture, where inflectional rules are assigned to specific lexical entries. BEESLEY's (2001) inflectional dictionary of Arabic is an example of this architecture (Fig. 10).

Lemma	Underlying form	Surface form
banay "build"	*banayat*	*banat* "(she) built"
qawul "say"	*qawula*	*qaala* "(he) said"

Fig. 10. Conjugation in Arabic through rule application (BEESLEY, 2001)

Maintenance is the Achilles' heel of this architecture: the encoding of a lexical entry is not independent of others. The insertion or update of an entry may involve revising a rule, and such revision may *a priori* cause effects on any entry in the dictionary.

Inflection in Arabic may be implemented in a dictionary with lemma-based architecture instead, as NEME (2011) shows for verbs, and NEME, LAPORTE (forthcoming) for nouns.

Conclusion

In this article, we wondered what features of an NLP dictionary make it easy to handle, and we focused on the human tasks involving such dictionaries: construction, update, check and correction.

The Lexicon-Grammar style of encoding dictionaries facilitates human tasks by prioritizing readability in its organization. The first results obtained with the aid of this method were published in 1968. Paradoxically, more recent international recommendations and projects have consistently overlooked these aspects. Since the elaboration of recommendations and standards systematically attempts to take into account the point of view of all relevant actors in the process, we are bound to conclude that NLP dictionary authors fail to voice their needs…

Authors of dictionaries of the world, unite!

Authors of NLP dictionaries, it is your own interest to practice and test existing encoding systems and to select those that best meet your needs.

Cited works

BEESLEY, K.R. 2001. "Finite-State Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plans in 2001". *ACL/EACL Workshop 'Arabic Language Processing: Status and Prospects'*, pages 1-8.

BOONS, J.-P.; GUILLET, A.; LECLÈRE, Ch. 1976. *La structure des phrases simples en français. 1 : Constructions intransitives*. Genève : Droz.

CALZOLARI, N.; MCNAUGHT, J.; ZAMPOLLI, A. 1996. *The EAGLES Guidelines*, Available at: http://www.ilc.cnr.it/EAGLES/browse.html Last access on 06-07-2012.

CONSTANT, M.; TOLONE, E. 2010. "A generic tool to generate a lexicon for NLP from Lexicon-Grammar tables". Michele De Gioia (org.), *Actes du 27e Colloque international sur le lexique et la grammaire (L'Aquila, 10-13 septembre 2008), Seconde*

partie, volume 1 of Lingue d'Europa e del Mediterraneo, Grammatica comparata. Rome: Aracne, pages 79-193.

COURTOIS B. 1990. « Un système de dictionnaires électroniques pour les mots simples du français », *Langue française* 87:11-22, Paris : Larousse.

FELLBAUM, Ch. (ed.). 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

FILLMORE Ch.; ATKINS S. 1994. "Starting where the dictionaries stop: The challenge for computational lexicography", S. Atkins, A. Zampolli (eds.), *Computational Approaches to the Lexicon*. Oxford University Press, pages 349-393.

FRANCOPOULO G.; GEORGE M.; CALZOLARI N.; MONACHINI M.; BEL N.; PET M.; SORIA C. 2006. "Lexical Markup Framework (LMF)", *LREC*, pages 233-236.
GRISHMAN R., MACLEOD C., MEYERS A. 1994. "COMLEX Syntax: Building a Computational Lexicon", *Coling*, pages 268-272.

GROSS, M. 1975. *Méthodes en syntaxe. Régime des constructions complétives*, Paris: Hermann, 414 pages.

GROSS, M. 1984. "Lexicon-grammar and the syntactic analysis of French", *COLING and ACL*, pages 275-282.

KIPPER, K.; KORHONEN, A.; RYANT, N.; AND PALMER, M. 2006. "Extending VerbNet with Novel Verb Classes". *LREC*, pages 1027-1032.

LAPORTE, É.; TOLONE, E.; CONSTANT, M. 2013. "Conversion of Lexicon-Grammar tables to LMF: application to French". Francopoulo, George (eds.), *LMF: Lexical Markup Framework, theory and practice*, Hermès/ISTE/Wiley.

MILLER, G.A. (ed.). 1990. WordNet: An on-line lexical database [Special Issue]. *International Journal of Lexicography* 3:235–312.

MUNIZ, M.C.M.; NUNES, M.G.V.; LAPORTE, É. 2005. "UNITEX-PB, a set of flexible language resources for Brazilian Portuguese". *Workshop on Technology on Information and Human Language* (TIL'05), pages 2059–2068.

NEME, A. 2011. "A lexicon of Arabic verbs constructed on the basis of Semitic taxonomy and using finite-state transducers". *International Workshop on Lexical Resources* (WoLeR) at ESSLLI.

NEME, A.A.; LAPORTE, É. (forthcoming). "Pattern-and-root inflectional morphology: the Arabic broken plural". 35 pages.

SAGOT, B.; CLEMENT, L.; VILLEMONTE DE LA CLERGERIE, É.; BOULLIER, P. 2006. "The Le*fff* 2 syntactic lexicon for French: architecture, acquisition, use". *LREC*, pages 1348-1351.

TOLONE, E. 2012. *Analyse syntaxique à l'aide des tables du Lexique-Grammaire français*. Éditions Universitaires Européenes, 352 pages.

VAN DEN EYNDE, K.; MERTENS, P. 2003. « La valence: l'approche pronominale et son application au lexique verbal ». *Journal of French Language Studies* 13, pages 63-104.