CONVERTING LARGE ON-LINE VALENCY DICTIONARIES FOR NLP APPLICATIONS: FROM PROTON DESCRIPTIONS TO METAL FRAMES

GEERT ADRIAENS [1,2] GERT DE BRAEKELEER [1]

[1] Siemens-Nixdorf Software Center Liège
Rue Des Fories 2, 4020 Liège, Belgium
[2] Leuven University Center for Computational Linguistics
Maria-Theresiastraat 21 3000 Leuven, Belgium
geert@et.kuleuven.ac.be

0. Abstract

In this paper, we report on a large-scale conversion experiment with on-line valency dictionaries. A linguistically motivated valency dictionary in Prolog is converted into a valency dictionary for a large-scale machine translation system. Several aspects of the two dictionaries and their background projects are discussed, as well as the way their representations are mapped. The results of the conversion are looked at from an economic perspective (fast coding for NLP), and also from a computational-lexicographic perspective (requirements for conversions and for standardization of lexicon information).

1. Introduction

One of the major bottlenecks for large-scale NLP applications such as the METAL® MT system[1] is the acquisition of their lexicons[2]. Whereas the development and fine-tuning of the grammars of such systems reaches its saturation point after a few years of R&D, the extension of their lexicons is a constant and ever growing concern. In order to speed up the lexical acquisition process, coding tools are developed to increase the human lexicographer's productivity and existing electronic dictionaries are looked for that can be converted and integrated with the particular NLP application at hand.

In this paper we report on a large-scale conversion effort with an eye to enhancing the METAL verb dictionaries with several thousands of entries. While the system is capable of defaulting the necessary morphological information for verbs on the basis of their surface appearance (cp. Adriaens & Lemmens 1990), it cannot automatically create the complex syntactic-semantic valency information, i.e. the quantitative and qualitative characterization of the arguments of a verb. Still, this information is of crucial importance for the system to parse and translate correctly. Valency characterizations can be used to discriminate different readings of a sentence during analysis (cp. e.g. the different usages of *hail*: *it is hailing, she hailed curses at me, he hailed me from the window, the people hailed him king*). Moreover, they are often useful for disambiguating purposes with an eye to translation: for Dutch, for instance, *to reach for something* is a usage that needs a different translation from *to reach somebody something* (*pakken* versus *overhandigen*). (For a detailed discussion of the importance of valency for NLP and MT in particular, we refer to Gebruers 1991.) To recognize the need for detailed valency descriptions in NLP applications is one thing, to acquire them is less self-evident. In a system like METAL, the valency feature on verbs represents the most complex and hard-to-code element in its lexical representations. Hence, to automate and speed up the acquisition process, we used electronic valency dictionaries for Dutch and French as coded by the PROTON project (see van den Eynde et al. 1988, Eggermont & van den Eynde 1990, Eggermont et al. forthcoming) as our starting point. The conversion was a non-trivial exercise in computational lexicography for several reasons. First, the PROTON databases are mainly descriptive and exhaustive in nature; they were not conceived with particular NLP applications in mind. METAL, on the other hand, seeks parsimony for efficient computational treatment within a machine translation application. More in particular, PROTON codes one entry per valency frame of a verb, whereas METAL merges valency patterns into "superframes", storing these only once for each verb. Second, their representation formalism is based on a particular distributional linguistic approach (the Pronominal Approach, see 2.2) not completely alien to the METAL representation, but not straightforwardly convertible either. And third, the PROTON databases take the form of Prolog clauses, whereas METAL uses Lisp lists. Beside the purely practical goal of fast lexicon extension, there are a few interesting questions to be asked that may be relevant beyond that goal:

- Is such a conversion worth the effort of defining a "waterproof" mapping between the source and target formalisms, and of developing the programs to do the mapping? In other words, could we not simply have coded the several thousand verbs by hand instead of spending months on the conversion?

- To what extent are these conversion experiments useful for an attempt at defining a theory-neutral standard for the representation of valency information in verb dictionaries for NLP applications? Or, less ambitiously, can we come up with a set of requirements for convertibility of lexical resources?

2. Verbal valency descriptions

2.1 General considerations

[1] Metal® is a Siemens-Nixdorf (SNI) product. The University of Leuven co-develops the Dutch-French, French-Dutch and French-English language pairs with SNI.

[2] Cp. Walker, Zampolli & Calzolari forthcoming, Boguraev & Briscoe 1989, Zernik 1989.

In linguistic terms, verbal valency can be characterized as lexically controlled structural potential of a verb; in artificial intelligence terms, one would say that the verb has a frame structure with different role slots to be filled by constituents in the sentence. Since the verb is often the nucleus of information around which the different sentential elements are organized, it is important for an NLP system to contain this valency information. What then are the aspects of representation one has to take into account, in particular with an eye to NLP applications? The first problem to be solved is what falls within the scope of the verb's valency (i.e. the number and kind of valency-bound elements) and what falls outside of it (i.e. the free adjuncts of the sentence). An answer to this question leads to a quantitative classification of verbs as monovalent (only one valency element), bivalent (two) etc, and a qualitative classification of verbs as intransitive (subject, no object), transitive (subject and object), etc. Next, one faces the problem of the distinction between obligatory and optional valency-bound elements (a distinction that is of particular importance to a role assignment algorithm). And finally, one must name, categorize and subcategorize these elements, defining legal fillers for a certain slot. If a verb has several valencies (corresponding to different syntactic/semantic readings), an additional representational matter to be handled (at a higher level of lexicon organization) is the way to store the different valencies. Are patterns stored separately with a repetition of the verb for each pattern? Can patterns be collapsed and stored just once with the verb? Decisions on these matters influence the database organization and consultation for NLP applications. In the next two subsections, we will show how the two formalisms between which the conversion was made try to provide answers to the representation questions formulated here, in particular for large sets of French and Dutch verbs.

2.2 PROTON

2.2.1 The PROTON project

The Proton (*Prolog en taalonderzoek, Prolog and linguistic research*) project started in 1986 with as one of its major objectives the construction of on-line valency dictionaries for French and Dutch. The starting point was not a particular NLP application, but rather a linguistic concern for descriptive correctness and completeness. Still, computational concerns were present right from the start, which led to the choice of Prolog as the declarative language for storing and processing the verbs (with processing ranging from simple retrieval of specific subsets of verbs to NLP applications in computer-aided language learning and parsing). Paper dictionaries, both general (Le Petit Robert for French, Van Dale Basiswoordenboek for Dutch) and valency dictionaries (Busse & Dubost 1983 for French) were used as background material. For the actual coding of the verbs, a particular distributional framework formed the basis, viz. the Pronominal Approach[3]. Although there are many interesting sides to this approach (e.g. the exact methodology followed to determine reading distinctions in verbs), we are mainly interested here in the actual output of the lexicographic work, both quantitatively and for representation issues. As far as numbers are concerned, the current status of the valency dictionaries of Dutch and French is the following. The Dutch valency dictionary contains about 4500 verbs; since each syntactic/semantic reading is coded separately, there are actually about 6300 valency patterns coded. For French the two figures are 4000 and 8500[4]. (Note, in the passing that the frame/verb ratio is 1.3 for Dutch and 2.1 for French.) A rough estimate of the effort spent in doing this coding is 2 man-years for French, 1 man-year for Dutch. The difference is mainly due to the fact that French was the first language Proton started out with; by the time Dutch was handled, coding experience and coding tools were available.

2.2.2 The PROTON valency representation

Proton database entries are Prolog facts, consisting of a three-place *v* predicate; the three arguments are an identification number, the verb's infinitive, and a list structure containing the information related to one valency realization. Due to space limitations, we have to refer to De Braekeleer 1991 for a formal account of this list structure; for examples, we refer the reader to section 4. For clarity's sake, we informally give the meaning of important abbreviated notions: *p0* relates to the notion of *subject*, *p1* to that of *direct object*, *p2* to *indirect object*, *p3* to a specific *prepositional object* with *de* (related to French *en*), *pprep* to other *prepositional objects*, *ploc/pmanner/ptemp/pqt* to *adverbial of location/manner/time/quantity* respectively.

In general, it can be said that Proton valency entries are dense in information, but on the other hand somewhat loosely structured. We will see below that NLP applications like Metal have a more rigid structure that is not so dense in information. For a conversion experiment this difference is both an advantage and a disadvantage: the advantage is that one can go from structures that contain more than one needs; the disadvantage is that the determination of what maps to what is not straightforward.

2.3 METAL

2.3.1. The METAL system

In contrast to Proton, Metal is a specific NLP application, viz. a machine translation system. Its German-English, English-German, German-Spanish, Dutch-French and French-Dutch systems are commercially available; French-English, German-Danish, English-Spanish, Spanish-English and Russian-German are under development. Full descriptions of the system can be found elsewhere[5]. A brief account of

[3] See Blanche-Benveniste et al 1984 or Eggermont et al 1990 for full accounts of the Pronominal Approach.

[4] In the course of 1991 the French valency database will be commercially available in electronic form (Eggermont et al. forthcoming).

[5] See Bennett & Slocum 1988, Thurmair 1989, Adriaens & Caeyers 1990 for general overviews; a general description of the lexicon format can be found in Adriaens & Lemmens 1990.

valency in Metal can be found in Gebruers 1988; an in-depth study of valency and machine translation bringing together work in the Proton and Metal projects is the topic of Gebruers 1991. Here, we will just give a general idea of the place of valency information in the Metal system and of how this information is used in the translation process. Valency patterns are stored as a feature-value pair on verbs in the monolingual dictionaries, in such a way that all patterns are coded only once with the verb; reading distinctions can give rise to different valency patterns, but even then they are all stored together with the verb. During analysis by an augmented context-free grammar (handled by a chart parser), rules at sentence level call a procedure for role assignment to the constituents of the sentence. This process is an intricate combination of general pattern matching algorithms and linguistically defined procedures (triggered by the valency information) for determining the best fitting valency pattern. In fact, the role assignment process can be said to consist of a *grammar within the grammar, and a parser within the parser*; it takes up a substantial proportion of the total time spent on sentence analysis. During transfer, valency information is again used (in the transfer dictionary) to disambiguate among different verb readings. For mapping into the target language, there are two approaches within Metal that have implications for the amount of valency-related information in the transfer dictionary. One approach tries to build a minimal hypothetical target language frame on the basis of the source role assignments and some crucial mapping information (e.g. for *to like -> plaire*, the subject is mapped into an indirect object, and the direct object becomes the subject: *I like you -> Tu me plais*). It then searches the monolingual target dictionary for a valency pattern that fits best with its hypothesis. The other approach tries to build the target frame without using the target dictionary at all: on the basis of the source role assignments and mapping information in the transfer dictionary, it builds the valency information for the target (see Gebruers 1991 for a detailed comparison of these approaches, with their advantages and disadvantages). In short, valency plays an important role in all phases of the translation process[6], involving complicated grammar and coding work. We conclude this brief sketch of valency in Metal by adding some figures of the size of the monolingual dictionaries. At the time of the conversion (March 1990), Metal contained 1600 Dutch verbs with 2050 valency patterns (a frame/verb ratio of 1.3) and 1055 French verbs with 1600 patterns (a frame/verb ratio of 1.5). Let us add right away that partly thanks to the conversion effort we were able to increase these figures drastically in a short period of time (see section 4). Currently, there are 3000 Dutch verbs with 3700 valency patterns (frame/verb ratio = 1.2) and 2130 French verbs with 2850 valency patterns (frame/verb ratio = 1.3). In general, all other monolingual dictionaries of the commercially available systems (i.e. English, Spanish and German) also contain over 2000 verbs (2500, 2300 and 4000 respectively).

[6] See Gebruers 1991, 206-221 for an overview of valency treatment in other MT systems (TAUM, SUSY, GETA-ARIANE, VAPRE, EUROTRA).

2.3.2 The METAL valency representation

In METAL, valency is coded as one of the feature-value pairs on the lexicon entries for verbs (along with other information about morphology, syntax and semantics). Since the system is written in Lisp, its elements show the typical Lisp list structure. As for Proton, we have to refer to De Braekeleer 1991 for a full formal account of the METAL valency format; examples can be found in section 4. The meaning of some important abbreviations is the following: *$SUBJ* stands for *subject*, *$DOBJ* for *direct object*, *$IOBJ* for *indirect object*, *$ADV* for *adverbial complement*, *$POBJ* for *prepositional object*, *$SCOMP* for *subject complement*, and *$OCOMP* for *object complement*. *N1*, *N0*, *IMPS* and *ADJ* indicate *nominal*, *sentential*, *impersonal* and *adjectival* subcategorizations respectively. Adverbial complements are further divided into *LOC(ative)*, *MAN(ner)*, *MOV(ement)*, *R(a)NG(e)*, *T(e)MP(oral)* and *MEA(sure)*. Further subcategorization information is rendered as feature-value pairs, e.g. *(TYPE P1)* roughly corresponds to +human role fillers. Metal further uses the "OPT" atom in its valency patterns to indicate the optional valency-bound elements. Obligatory elements come first, those following the "OPT" atom are optional. Finally, the valency pattern contains General Frame Tests (after the "GFT" atom). These tests are executed before the role assigning mechanism tries to find fillers; they concern features that if present at the clause level should have specific values: the auxiliary (values are *H/Z*, *hebben/zijn* for Dutch; *A/E avoir/être* for French) and the sentence's voice (VC; *A/P*, *active/passive*). It is interesting to note how in an application like Metal this kind of information (also present in the Proton descriptions) receives a special status with an eye to an efficient role assignment algorithm: if a valency pattern can be found not to apply because some restriction at the clause level is not satisfied, the pattern is discarded and no computation is wasted on checking the potential role fillers.

3. Mapping PROTON to METAL

It was already noted in 2.2.2 that the different origin of the two formalisms accounts for certain differences between them. Proton codes in an application-neutral fashion, exhaustively (aiming at descriptive adequacy), on a one-entry one-pattern basis, and in a relatively free format. Metal codes with an eye to a specific application (MT), pragmatically (what do we need for the application to run?), on a one-entry all-patterns basis (even collapsing some patterns in a superframe), and in a relatively rigid format easily digestible by software and lingware. Since the goal of the conversion was to derive the information needed in Metal, a first step was to link all the Metal specifications to the corresponding Proton ones. Given the detailed nature of the Proton valency schemes, there were very few gaps in this mapping. One is worth mentioning, though. Proton does not go as far as Metal in the subcategorization of the adverbial complements (Metal's $ADVs); range and movement complements are not treated in a consistent way. Below, we show part of the resulting mapping table (not all subcategorization details are shown; see De Braekeleer 1991, 61-62). It organizes the valency information from the Metal point of view: the relevant items are

optionality, naming of roles, categorization, subcategorization and general frametests.

	Metal	Proton Dutch	Proton French
optionality	OPT	[[] ...]	[[] ...]
roles	$SUBJ	p0	p0
	$DOBJ	p1	p1
	$IOBJ	p2	p2("lui")
	$POBJ	pprep	p2("y"), p3, pprep
	$SCOMP	(these two must be derived from	
	$OCOMP	several elements combined)	
	$ADV	advtype	advtype
categories	N1	cf. type	
	N0	cf. FCP / ICP	
	IMPS	p(p0,['_t'])	p(p0,['il'])
	ADJ	related_paradigms	

subcategorizations

ADVTYPE :	LOC	ploc
	TMP	ptemp
	MEA	pqt
	MAN	pmanner
	RNG, MOV	-----

TYPE :	P1	"wie"	"qui"
	P0	"wat"	"que", "quoi"

general frametests:

VC:	A	related_par.	p(reform,['passif être'])
	P	related_par.	absence of above

AUX Dutch	Z	p(reform,['zijn+vd.', ...])
	H	p(reform,['perfectum hebben', ...])
French	A	auxiliary(['avoir'])
	E	auxiliary(['être'])

4. Aspects of the conversion software

Ideally, the conversion should be a fully automatic process that takes the Proton database as input and delivers a Metal monolingual verb lexicon. Given that the Proton database also contains a field with several translations for each verb reading, we could even envisage creating transfer entries for the verbs as well. Yet, there are several reasons why we could only achieve a semi-automatic conversion. As to the automatic generation of transfer entries, this idea had to be abandoned altogether, because it was too hard to pinpoint the distinctive information among the different patterns and translate that into contextual tests and actions in the Metal transfer dictionaries. Still, the translation field was preserved in the conversion output, so that lexicographers coding the transfer entries already had the translations on-line. As to the fully automatic generation of a monolingual lexicon, several problems could not be overcome. First, we already noted in the previous section that not all information needed for Metal was present in the Proton database; this implies that manual checks for completeness of the frames had to be made in any case. Second (and most important), we could find no satisfactory algorithmic solution to the problem of mapping the one-entry one-valency-pattern organization of Proton into the one-entry all-patterns organization of Metal. Note that this is not a simple matter of collecting all the separately coded valency patterns for the same verb, and storing them once as a long list with the verb in the target database. For one thing, Metal does not need all possible valency patterns for its purpose of machine translation; the amount of patterns is kept as small as possible for efficient storage and computation reasons. Moreover, the patterns that remain are merged into "superpatterns" or "superframes" as much as possible; where relevant for translation, the transfer dictionaries take them apart again. The way Metal lexicographers decide on distinguishing valency patterns (verb readings) monolingually proved hard to translate into a foolproof algorithm; there are at the most some intuitions, heuristics or rules of thumb. Hence, it was decided to convert on a per pattern basis, and leave the merging of patterns to the human lexicographer.

The conversion software itself is written in Common Lisp (about 1000 lines of code). It works in two phases. First, the Proton Prolog clauses pass through a finite-state transducer interpreting them as plain character strings. The output of this pass is "lispified Prolog": Prolog clauses are turned into Lisp lists. At the same time, the necessary conversions at the character level are taken care of: characters that would have a special meaning to the "Lisp reader" software (such as a comma or a backquote) are neutralized, and the extended ASCII-character sequences for accented characters are turned into Metal's ISO-8859-1 characters. The second pass parses the lists and converts them into structures whose most important field is the Metal frame. Additional software takes care of putting the Metal frames in their canonical order (i.e. a subject is coded before an object, etc.), and provides tools for lexicographers to manipulate the conversion output. As an illustration, we give one simple example of what the input and the output of the conversion look like:

Proton input:

```
v(24720,'dégager',
[ex('r : dégager qqn d'une charge'),
tr(['ontslaan (van)','ontheffen (van)']),
p(p0,[je,nous,on,qui,elle,il,ils,'celui-ci','ceux-ci']),
p(p1,[te,vous,'se réc.','l'un l'autre','se réfl.',
      qui,la,le,les,'en Q','celui-ci','ceux-ci']),
p(p3,[en,'en(de_inf)',quoi,'celui-ci','ceux-
      ci','ça','ça(de_inf)']),
p(reform,['passif être']), pivot(p1,p0,de_inf,p3)]).
```

Metal output:

```
dégager
Example : (r : dégager qqn d'une charge)
Transfer : (ontslaan (van) ontheffen (van))
Proton : ((reform passif être)
          (p3 en en(de_inf) quoi celui-ci ceux-ci ça ça(de_inf))
          (p1 te vous se réc. l'un l'autre se réfl. qui la le
              les en Q celui-ci ceux-ci)
          (p0 je nous on qui elle il ils celui-ci ceux-ci))
(($SUBJ N1 (TYPE P1))
 ($DOBJ N1 (TYPE P1) (PRN RFX))
 ($POBJ N1 (PREP de) N0 (ICP de) (PIV
$DOBJ)))
```

5. Discussion of results

Using the conversion software, the complete Proton database (at that time, i.e. March 1990, consisting of 8500 valency structures for French and 6000 for Dutch)

was processed into a database with Metal valency patterns that could form the basis of manual coding. In the first place, checks were run to compare the results of the conversion with the frames already coded in the dictionary. This already led to an improvement of the existing database. In the second place, additional verb coding is now being done on the basis of the conversion output, and not from scratch (i.e. from paper dictionaries).

The total effort spent on developing the software (including the preliminary study phase constructing the mapping table) was about four man-months. When we compared the time needed to code Metal valency frames starting from scratch (the way the first 1000 verbs were added to the system) with the time needed to code frames starting from the output of the conversion, we found that on the whole, and subtracting the conversion development effort, coding productivity is speeded up by a factor of 2. In other words, the practical goal of fast extension of the verb dictionaries was certainly reached.

As to the more general questions of requirements for convertibility of lexical resources or standardization of lexical information, a few remarks are in place. First, in our case the input lexical resource was in a fairly easily convertible format, viz. Prolog clauses. Even so, since it was the first time the Proton databases were used outside of the project, several ambiguities and inconsistencies were found that needed correction before the conversion could take place. A basic requirement for convertibility then seems to be a rigid description of the syntax and semantics of the database entries; before the resource is made available to the outside world, it should be checked thoroughly against its own specifications (parsers can be generated automatically on the basis of a BNF-like syntax). More ambitiously, if the formats of valency information in different applications were known, the resource could be made available along with converters or converter specifications. As to the long-term goal of standardization, we are planning to use the experiences gathered from the conversion (along with knowledge about other formalisms, like that used in EUROTRA or in the databases of the Nijmegen Centre for Lexical Information CELEX) to study the requirements for a theory-neutral and application-neutral standard for valency representation. Since valency is not restricted to verbs, but also concerns adjectives and nouns, the standard could even try to be category-neutral as well.

Although the Proton-Metal conversion proved a successful experiment in computational lexicography, many linguistic and computational issues concerning valency and its processing have not been touched upon here and certainly need further research. To name but a few: nominal and adjectival valency, a foolproof methodology for making and/or merging reading distinctions, valency and idiomatic expressions, the interactions of the different types of valency information in an NLP application, and the link with more semantically oriented approaches to valency. On the basis of the availability of large amounts of valency data, and the experience with different formalisms, we hope to be able to tackle some of these issues in the future.

References

Adriaens, G. & H. Caeyers (1990) -- "Het automatisch-vertaalsysteem METAL : van onderzoek tot commercieel produkt" in *Ingénieur & Industrie*, Dec 1990, 281-288.

Adriaens, G. & M. Lemmens (1990) -- The self-extending lexicon: off-line and on-line defaulting of lexical information in the METAL translation system. In Proceedings of the 13th COLING, Vol 3, 305-307.

Bennett, S. & J. Slocum (1988) - The LRC Machine Translation System. In J. Slocum (ed) Machine Translation Systems Studies in Natural Language Processing. Cambridge: Cambridge University Press, 111-140.

Blanche-Benveniste et al. (1984) -- Pronom et Syntaxe, l'approche pronominal et son application au français. Paris: Selaf.

Boguraev, B. & E. Briscoe (eds) (1989) -- Computational lexicography for natural language processing. London: Longman.

Busse, W. & J.P. Dubost (1983) -- Französisches Verblexicon. Die Konstruktion der Verben im Französischen. Stuttgart: Ernest Klett.

De Braekeleer, G.(1991) -- De conversie van PROTON valentiestructuren naar METAL frames : conversie of (contr)aversie. University of Leuven Master's Thesis in Computational Linguistics.

Eggermont C., E. Broeders & K. van den Eynde (forthcoming) -- Dictionnaire automatisé des valences des verbes français. University of Leuven.

Eggermont, C. & K. van den Eynde (1990) - A prono-minal basis for computer-assisted translation: the Proton project. In Thelen, Lewandowska & Thomaszczyk (eds), Translation and meaning I. Maastricht: Euroterm, 1-14.

Gebruers, R. (1988) -- Valency and MT: recent developments in the METAL system. In Proceedings of the 2nd ACL, Austin, 168-175.

Gebruers, R. (1991) -- On valency and transfer-based machine translation. An inquiry into the language-technological applicability of theoretical valency concepts. University of Leuven PhD Thesis.

Thurmair, G. (1990) -- Aufgabentyp Linguistik: Projekt METAL. In D. Nebendahl (ed), Expertensysteme Teil 2: Erfahrungen aus der Praxis. Siemens AG, München.

van den Eynde, K. et al. (1988) -- The pronominal approach in NLP: A pronominal feature analysis of coordination in French. In *Computers and Translation* 3, 177-213.

Walker, D., A. Zampolli & N. Calzolari (eds) (forthcoming) -- Automating the Lexicon: Research and Practice in a Multilingual Environment. Oxford: Oxford University Press.

Zernik, U. (1989) -- Paradigms in lexical acquisition. In Proceedings of the first international lexical acquisition workshop, Detroit (Zernik ed.).