Using a rich external valence dictionary with an implemented XLE/LFG grammar

Agnieszka Patejuk
aep@ipipan.waw.pl

Introduction

This paper discusses how Walenty, an innovative valence dictionary of Polish, was used in an LFG grammar of Polish implemented in XLE. It begins with introducing the distinctive features of Walenty that make it attractive from the perspective of use in an implemented LFG grammar, then it proceeds to presenting the procedure of interpreting and converting valence specifications from Walenty to LFG formalism, focusing on the following issues: assignment of the grammatical function to arguments, taking unlike category coordination into account and imposing the relevant constraints using available LFG mechanisms, including the issue of structural case assignment in Polish, the handling of passive voice and the issue of the analysis of optional arguments.

Walenty: an innovative valence dictionary of Polish

Walenty (Przepiórkowski *et al.* 2014b) is currently the largest and most precise valence dictionary of Polish – at the moment of writing it contains over 83 000 schemata for 15 000 lemmata. Unlike many other dictionaries, it contains not only schemata for verbs, but also for nouns, adjectives and adverbs. For reasons of space, this paper focuses on verbal schemata exclusively: there are over 63 000 schemata for 12 000 lemmata, which gives 5.25 schemata per lemma on average. More importantly, Walenty has a number of interesting if not innovative features.

First, it assumes that a syntactic schema (see (1) and (3)) consists of argument positions that are modelled as sets of categories that can realise a given position – the contents of such sets are specified according to the coordination test of Szupryczyńska 1996: if two or more categories can be coordinated within one position (say, a noun phrase and a clause as the subject), then it is a multi-element set. This feature is innovative in that it explicitly accounts for the coordination of unlike categories – other valence dictionaries would use separate valence schemata for different categories (as in (4)): one with a nominal subject and another one with a clausal subject, which results in either an XOR (exclusive OR) specification of the subject – either a nominal phrase or a clause, so that the possibility of having a coordinated unlike category subject is ruled out, or it might be interpreted as an OR specification, allowing for such coordination at the cost of overgenerating (allowing such coordination when it is not possible). It is possible to avoid such problems by adopting the solution proposed in Walenty, where syntactic positions are modelled as sets which correspond to an OR specification, which means that the given position can be filled by any single set element (only nominal or only clausal) or by any combination of these elements, which accounts for unlike category coordination. If the given position can be filled in more than one way but the relevant elements cannot be coordinated, the XOR specification is obtained by creating separate schemata with singleton sets corresponding to the relevant argument. Secondly, it explicitly identifies the subject position (`subj`) – understood as the argument that drives verbal agreement, regardless of its category, so it takes into account non-canonical subjects – and the object (`obj`) – defined as the argument which can become the subject under passive voice, so the passivisable object, regardless of its category (and case marking, if it happens to be a nominal); it must be highlighted that Walenty does not distinguish other grammatical functions than these two – other sets are not labelled with any grammatical function. Walenty provides information about the fact that an argument requires structural case – together with information about the grammatical function of the argument and information about the syntactic context, this makes it possible for the grammar to resolve the appropriate value of case. Furthermore, Walenty explicitly accounts for raising and control by using `controller` and `controllee` labels to establish relations between respective arguments (see (6)). These labels are also used to mark agreement with predicative arguments – the argument that controls agreement is marked as `controller`, while the argument whose features it inherits (possibly embedded in a prepositional phrase) is marked as `controllee` (see (7)). Next, Walenty introduces a new class of `xp` arguments – defined by their semantics rather than category: these include ablative, adlative, locative (see (1)), etc. arguments. For each type of `xp`, there is a defined list of its realisations (see (2)), which results in economic, readable and coherent schemata. While using a plain locative `xp(locat)` means that all its realisations are possible with a given schema, it is also possible to use the subtype mechanism – a list restricting the original realisation list to specified ones, which ensures accuracy if a given schema requires extra constraints. Finally, Walenty is one of the few valence dictionaries that include a rich phraseological component (Przepiórkowski *et al.* 2014a) – it explicitly specifies lexicalised arguments and constraints imposed on them, with the possibility of embedding such constraints arbitrarily deep (see the paper for an example).

Since Walenty uses its own formalism, it is not tied by the constraints of any particular grammar formalism and it can be used with any grammar formalism (so far, this has been done for LFG, DCG and CG), provided that the grammar writer is able to interpret the specifications provided in Walenty and convert them to the particular formalism. The aim of this paper is to show an idea of how this can be done for LFG – on the basis of selected

phenomena. Furthermore, it is believed that Walenty is a good example of a complex, rich valence dictionary, whose adoption can be of great use for the purposes of parsing.

Interpreting Walenty and converting it to LFG

Due to the amount of data to be processed, the conversion is done automatically using a Python script; this method also ensures consistency and coherence. Let us first introduce the basics of converting Walenty schemata to LFG constraints, and then proceed to the discussion of details of selected phenomena.

The general idea of converting a valence schema to LFG constraints is very simple: each argument must be assigned a grammatical function and then appropriate constraints relevant to this argument must be imposed. Since the grammatical function must be chosen to apply relevant constraints, let us start with the procedure of **choosing the grammatical function**. As mentioned above, only two grammatical functions are already specified in Walenty: the subject, defined as the argument that drives verbal agreement (regardless of its category), and the object, defined as the argument that can become the subject under passive voice (again, regardless of its category and, consequently, case, if applicable). The remaining grammatical functions are not specified in the dictionary and must be assigned in the process of conversion – the following basic specification is used:

- OBJ$_\theta$: thematic/secondary object – nominal, it does not passivise,
- OBL (oblique): non-semantic prepositional phrase,
- OBL$_\theta$ (thematic oblique): semantic prepositional phrase,
- COMP: closed clausal complement,
- XCOMP: open infinitival complement,
- XCOMP-PRED (predicative complement): open predicative nominal/adjective (possibly embedded in a prepositional phrase)[1]

The specification outlined above works perfectly as long as the relevant argument position contains only one realisation (it is a singleton set in Walenty). If it is not the case (see (5)), the choice of the grammatical function becomes problematic as different categorial realisations of the relevant argument position may correspond to different grammatical functions. Typically, a coordinate phrase corresponds to one grammatical function in f-structure, so a common grammatical function should be chosen: if a nominal phrase would normally be assigned the OBJ$_\theta$ grammatical function (because it does not passivise) and a clausal complement the COMP grammatical function, which of these should be assigned to their coordination? This very problem has been discussed in LFG literature: Dalrymple and Lødrup 2000 suggest that it should be an object grammatical function (here: OBJ$_\theta$), while COMP should be treated as an elsewhere grammatical function where the nominal realisation is not possible. The conversion of Walenty is inspired by this solution – it uses the ranking of grammatical functions defined in (8) to choose the common grammatical function from the set of candidates: the conversion script assigns each realisation of the relevant argument position the corresponding ranking and then the highest ranked grammatical function candidate is chosen. According to the ranking in (8), if an argument position can be realised as a non-semantic prepositional phrase (OBL) or a non-passivisable nominal (OBJ$_\theta$), it would be assigned the OBL grammatical function. If a clause (COMP) can be coordinated with a non-passivisable nominal (OBJ$_\theta$), it would be one of the OBJ-<CASE> functions. The XCOMP and COMP are the lowest ranked grammatical functions: they are only chosen when the clause or infinitive are the only realisations in the set corresponding to the relevant argument position.

Once the grammatical function is chosen, one may proceed to **imposing relevant constraints**: the method of doing this depends on whether the given argument position corresponds to a singleton set (only one realisation) or not (it is an instance of unlike category coordination). In the former case the procedure of imposing constraints is simple as it involves using plain constraints, as opposed to off-path constraints required when unlike category coordination is possible. This is because, as discussed in Patejuk and Przepiórkowski 2012, an alternative of plain constraints such as in (9) is evaluated once and distributed to each conjunct under coordination, which makes it impossible for particular conjuncts to satisfy different disjuncts of such a constraint. The desired effect of evaluating the disjunction separately for each of the conjuncts under coordination can only be achieved using off-path constraints as in (10), which makes the relevant constraints more complicated and forces a particular way of imposing constraints in implemented grammars. This is because in XLE off-path constraints are non-constructive – while there are constraining off-path equations, there are no defining ones (the latter type is, however, mentioned in theoretical works such as the recent second editions of Dalrymple 2001 and Bresnan 2000). Because of this limitation, the constraints imposed by the predicate must be formalised as constraining equations – also in the case of plain, non-off-path constraints, for the sake of consistency.

[1] As an alternative, the closed PREDLINK grammatical function could be used.

Let us illustrate this point by presenting the process of interpreting the information provided in Walenty and converting it to LFG constraints on the basis of **structural case assignment** – one of the most fundamental issues in Polish. Walenty marks the case of the relevant argument as `str`, which stands for "structural", which means that its realisation depends on a number of syntactic factors: the grammatical function of the argument (subject as opposed to object, passivisable or not), the presence of negation (when the argument is an object) and the part of speech of the verbal head imposing constraints. The valence dictionary provides information about the requirement of structural case and the grammatical function of the relevant argument, which is processed by the grammar, taking the syntactic context into account, in order to set the appropriate values of case. As discussed in Patejuk and Przepiórkowski 2014b for verbal heads in Polish, the structural object is marked for accusative case in the absence of negation and genitive case if negation is present – the proposed formalisation (see (11)) uses plain constraints, so it is not compatible with unlike category coordination. A formalisation of structural case assignment that does take this into consideration and uses off-path constraints is provided in Patejuk and Przepiórkowski 2014a (see (12)). It is worth mentioning that while it is possible to use templates in XLE to store fragments of plain constraints, templates can only be used for entire statements with off-path constraints, together with the plain part of the constraint to which the off-path constraint is attached (so, entire (12)) – it is not possible to assign fragments of off-path constraints to templates, which makes such constraints less economic and readable: it would be natural to assign the fragment handling structural case assignment to a nominal object (lines 2–5 in (12)) to a template and call it inside one of the off-path disjuncts when unlike category coordination is involved, only adding the disjunct related to the other categories (the last line in (12)), but this is not possible technically.

Another issue that is worth discussing is the method of handling **passive voice**: it is typically handled in grammars using a lexical rule, but an alternative method is used when converting Walenty – passive versions of schemata are created using the script. This is motivated by the fact that lexical rules only manipulate the value of the PRED attribute, which means that they change the grammatical functions (OBJ $\rightarrow$ SUBJ – the active object becomes the passive subject, SUBJ $\rightarrow$ OBL-AG/NULL – the active subject becomes the passive oblique agent or is dropped), but since this applies to the PRED attribute, it does not affect other constraints stored in the relevant lexical entry, such as control relations, which should also be changed accordingly – in case of control verbs such as TEACH, the control equation in the active is (↑ OBJ)=(↑ XCOMP SUBJ), whereby the object of TEACH is at the same time the subject of the infinitival complement of TEACH, while in the passive it should be changed to (↑ SUBJ)=(↑ XCOMP SUBJ) – because the active object becomes the passive subject. The use of a lexical rule will not introduce such a change and extra constraints must be added. It is, however, easy to introduce such changes in the process of conversion: when the passive version of the relevant schema is created, the script first changes the assignment of grammatical functions and then imposes the constraints, which results in changing all the appropriate constraints.

The next issue that must be considered when converting Walenty is the issue of **argument reduction**: by design, Walenty only provides maximal schemata (listing all possible arguments), but at the same time it assumes that all arguments are optional – in Polish most arguments may be dropped in the sense that they are not expressed. When performing the conversion, one must decide how to interpret this phenomenon: does the fact that the argument is unexpressed mean that it is not present in syntactic representation at all (it is reduced, it is not present in PRED) or is it the case that it is an implicit argument – it is not expressed lexically, but it is active syntactically (it is not reduced, it is present in PRED and it is filled with pro)? The proposed method of interpreting Walenty uses a hybrid solution – it divides arguments into two classes: obligatory (must be present in syntactic representation) and optional (can be removed from syntactic representation) on the basis of two criteria. First, if the absence of an argument changes the meaning of the predicate – as in the case of lexicalised arguments and the so-called reflexive marker SIĘ, which can be reflexive, reciprocal or inherent (in the last case it carries no semantic information, but it is required syntactically as in BAĆ SIĘ in (6), which means 'to fear', not 'to fear oneself') – then the argument is assumed to be obligatory and it must be lexical (overtly expressed). The second diagnostic is whether there is syntactic evidence that the relevant argument is syntactically active even though it has no surface realisation – the subject controls participial clauses and binds anaphora; also other arguments that serve as controllers – in this case the argument is obligatory, but it may be filled using an implicit pro argument. When none of these criteria is satisfied, the relevant argument is assumed to be optional and it may be reduced – this is done by removing it from PRED attribute and removing the respective constraints that apply to it. Removing arguments is such a way requires attention: controllers must not be removed unless the corresponding controllee is removed; however, once the controllee is removed, the `controller` label is removed from the controller and then it can also be reduced. An alternative approach to argument reduction would be to introduce implicit pro arguments for all arguments, but this would result in implicit clauses and prepositional phrases, which would introduce a lot of ambiguity – many predicates take both and a parse would be created for each such argument. Besides, there seems to be no syntactic evidence to support the introduction of such implicit arguments.

Conclusion

This paper briefly presented how valence information from Walenty, currently the biggest and most precise valence dictionary of Polish, can be used in an LFG grammar of Polish, presenting selected issues in more detail.

Figures (English translation of Walenty schemata is provided)

(1)
```
aklimatyzować się: subj{np(str)} + {xp(locat)}
acclimatise: subj{np(str)} + {xp(locat)}
```

(2)
```
xp(locat)-->                          xp(locat)-->
    advp(locat)                           advp(locat)
    [...]                                 [...]
    cp(int[gdzie])                        cp(int[where])
    [...]                                 [...]
    prepnp(koło,gen)                      prepnp(near,gen)
    prepnp(między,inst)                   prepnp(between,inst)
    prepnp(nad,inst)                      prepnp(above,inst)
    prepnp(na,loc)                        prepnp(on,loc)
    [...]                                 [...]
```

(3)
```
bawić: subj{np(str);cp(że)} + {np(str)}
amuse: subj{np(str);cp(że)} + {np(str)}
```

(4)
```
bawić: subj{np(str)} + {np(str)}
amuse: subj{np(str)} + {np(str)}

bawić: subj{cp(że)} + {np(str)}
amuse: subj{cp(że)} + {np(str)}
```

(5)
```
krzyczeć: subj{np(str)} + {np(str);cp(że);cp(żeby)} + {prepnp(do,gen)}
shout: subj{np(str)} + {np(str);cp(that);cp(to)} + {prepnp(to,gen)}
```

(6)
```
bać się: subj,controller{np(str)} + controllee{infp(_)}
fear (be afraid): subj,controller{np(str)} + controllee{infp(_)}
```

(7)
```
uchodzić: subj,controller{np(str)} + controllee{prepadjp(za,acc)}
pass (as): subj,controller{np(str)} + controllee{prepadjp(for,acc)}
```

(8)

#	GF
4	OBL-<SEM: ADL, DUR, LOCAT, MOD, PERL, TEMP>
3	OBL
2	OBJ-<CASE: DAT, GEN, INST, STR>
1	COMP, XCOMP

(9) $(\uparrow \text{GF ATTR1}) =_c \text{A} \vee (\uparrow \text{GF ATTR2}) =_c \text{B}$

(10) $(\uparrow \text{GF PRED})$
$(\leftarrow \text{ATTR1}) =_c \text{A} \vee (\leftarrow \text{ATTR2}) =_c \text{B}$

(11) $\text{STRCASE(GF)} \equiv [[\neg((\text{XCOMP}^* \uparrow) \text{NEG}) \wedge (\uparrow \text{GF CASE}) =_c \text{ACC}] \vee [((\text{XCOMP}^* \uparrow) \text{NEG}) =_c + \wedge [[(\uparrow \text{NEG}) =_c + \wedge (\uparrow \text{GF CASE}) =_c \text{GEN}] \vee [\neg(\uparrow \text{NEG}) \wedge (\uparrow \text{GF CASE}) \in_c \{\text{ACC, GEN}\}]]]]$

(12) $(\uparrow \text{GF PRED})$
$[\neg((\text{XCOMP}^* \text{ GF} \leftarrow) \text{NEG}) \wedge (\leftarrow \text{CASE}) =_c \text{ACC}] \vee [((\text{XCOMP}^* \text{ GF} \leftarrow) \text{NEG}) =_c + \wedge [[((\text{GF} \leftarrow) \text{NEG}) =_c + \wedge (\leftarrow \text{CASE}) =_c \text{GEN}] \vee [\neg((\text{GF} \leftarrow) \text{NEG}) \wedge (\leftarrow \text{CASE}) \in_c \{\text{ACC, GEN}\}]]] \vee (\leftarrow \text{CAT}) =_c \text{INF}$

References

Bresnan, J. (2000). *Lexical-Functional Syntax*. Blackwell Textbooks in Linguistics. Blackwell.

Butt, M. and King, T. H., editors (2014). *The Proceedings of the LFG'14 Conference*, Stanford, CA. CSLI Publications.

Dalrymple, M. (2001). *Lexical-Functional Grammar*. Academic Press.

Dalrymple, M. and Lødrup, H. (2000). The grammatical functions of complement clauses. In M. Butt and T. H. King, editors, *The Proceedings of the LFG'00 Conference*, University of California, Berkeley. CSLI Publications.

Patejuk, A. and Przepiórkowski, A. (2012). A comprehensive analysis of constituent coordination for grammar engineering. In *Proceedings of the 24rd International Conference on Computational Linguistics (COLING 2012)*.

Patejuk, A. and Przepiórkowski, A. (2014a). Control into selected conjuncts. In Butt and King (2014), pages 448–460.

Patejuk, A. and Przepiórkowski, A. (2014b). Structural case assignment to objects in Polish. In Butt and King (2014), pages 429–447.

Przepiórkowski, A., Hajnicz, E., Patejuk, A., and Woliński, M. (2014a). Extended phraseological information in a valence dictionary for NLP applications. In *Proceedings of the Workshop on Lexical and Grammatical Resources for Language Processing (LG-LP 2014)*, pages 83–91, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Przepiórkowski, A., Hajnicz, E., Patejuk, A., Woliński, M., Skwarski, F., and Świdziński, M. (2014b). Walenty: Towards a comprehensive valence dictionary of Polish. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, pages 2785–2792, Reykjavík, Iceland. ELRA.

Szupryczyńska, M. (1996). Problem pozycji składniowej. In K. Kallas, editor, *Polonistyka Toruńska Uniwersytetowi w 50. Rocznicę Utworzenia UMK. Językoznawstwo*, pages 135–144. Wydawnictwo Uniwersytetu Mikołaja Kopernika, Toruń.