Deploying the New Valency Dictionary Walenty in a DCG Parser of Polish

Marcin Woliński

Institute of Computer Science
Polish Academy of Sciences
E-mail: wolinski@ipipan.waw.pl

Abstract

This paper reports on developments in the Świgra parser related to the availability of the valency dictionary Walenty and their influence on the Składnica treebank of Polish. A method is proposed, which allows to use the rich valency data and yet to avoid unnecessary re-computation and reduplication of syntactic structures.[1]

1 Walenty – a valency dictionary of Polish

Walenty (Hajnicz et al., 2015; Przepiórkowski et al., 2014c,b,a) is a new comprehensive valency dictionary of Polish based on corpus data. Development of Walenty started by enhancing the dictionary created for Świgra parser (see below), but now the dictionary is much larger than the original and provides much richer information. In particular, the new dictionary includes not only verbs but also nouns, adjectives and adverbs. Walenty describes coordination of syntactically different arguments within a single syntactic position (so called unlike coordination), uses structural case (including partitive), provides semantic classification of some adverbial-like arguments (e.g., ablative and adlative), describes control and raising, and includes a rich phraseological component. Moreover, its syntactic level is being currently complemented with semantic frames.[2]

The following example depicts the syntactic schema of the verb CHCIEĆ 'want' used in the tree of Figure 1:

```
subj,controller{np(str)}
+controllee{np(str);cp(żeby);infp(_);advp(misc)}
```

[1]Work financed as part of the investment in the CLARIN-PL research infrastructure funded by the Polish Ministry of Science and Higher Education.

[2]The semantic level of Walenty is not yet used in Świgra, but it is a planned extension.

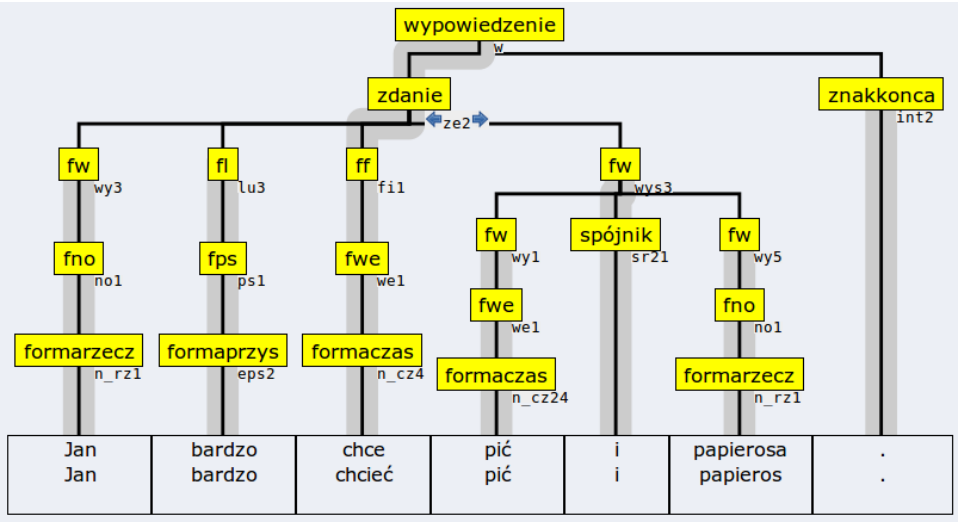

Figure 1: Parse tree for sentence (1)

According to Walenty a verb opens several syntactic positions which can be filled with specific arguments. The example schema comprises two syntactic positions (separated with a +). The first is marked as a subject realised by a nominal phrase in structural case `np(str)`. The second position specifies a list of argument types: a nominal phrase `np` in structural (in this case accusative or genitive depending on negation), a clause `cp` introduced by the complementizer `żeby`, an infinitival phrase `infp` in any aspect, an adverbial phrase `advp` of type `misc`. This notation means that the position can be filled by any of the listed arguments or by a coordination thereof.

Two positions are specially labelled: subject `subj` (the argument in this position influences morphological features of the verb) and passivable object `obj` (the argument in this position turns into a subject in passive voice). Other positions are unlabelled.

The two positions in the example are linked with a control relation expressed with the tags `controller` and `controllee`. By convention, control relations in Walenty are marked against positions, but it is understood that only some argument types take part in these relations. In this case the relation will hold between the argument filling the subject position and the subordinate clause or the infinitival complement.

In Walenty, due to the free word order of Polish, the order of positions within a schema and the order of argument types within a position is not important.

2 The treebank Składnica and the parser Świgra

Składnica (Woliński et al., 2011) is a treebank of Polish built on a 20,000 sentence subcorpus sampled from the manually annotated part of the National Corpus of Polish (Przepiórkowski et al., 2011). The primary format are constituency trees generated with the DCG (Pereira & Warren, 1980) parser Świgra (Woliński, 2004; Świdziński & Woliński, 2010) and then manually disambiguated and validated. The grammar stems from Świdziński's grammar (1992). Currently the treebank contains validated structures for 10,673 sentences.

Figure 1 shows Składnica-style annotation for the sentence:

(1) *Jan bardzo chce pić i papierosa.*
John much want to drink and cigarette

'John wants to drink and a cigarette very much.'

For terminals in the tree, the form and the lemma are shown. Internal nodes are represented by the name of the non-terminal category. But in fact each node carries several attributes specifying its syntactic features. (The attributes can be examined in the interactive interface of the parser.) One of these attributes is the type of an argument, as specified by Walenty.

In the example, the node for sentence (`zdanie`) consists of a 'required phrase' (`fw`, argument); a 'free phrase' (`fl`, adjunct); finite phrase (`ff`); and another `fw`. This last argument is a phrase featuring unlike coordination where a verbal phrase `fwe` in infinitive got coordinated with a nominal phrase `fno` in accusative, as allowed by the Walenty entry quoted in the previous section.

3 Deploying Walenty

3.1 Representation of valency schemata

Valency schemata given by Walenty are maximal in the sense that the dictionary does not list possible sub-schemata of a given schema. In Polish most of arguments are optional (in particular subjects are often omitted; see Section 3.3 for the full list of obligatory arguments in Walenty). Thus a method is needed to allow for the schemata to be realised partially in a controlled way.

One possible solution, used in the LFG grammar POLFIE (Patejuk, 2015), which also uses Walenty, is to compute all subsets of schemata in advance. Each subset leads to a separate lexical entry for the verb. This has the disadvantage of multiplying the lexical entries exponentially: a schema of length n has 2^n subsets (including the empty one).

Schema lengths in Walenty are listed in Table 1. The median of lengths in the dictionary is 3, which means a typical schema gets rewritten into 8 lexicon entries. Moreover, verbs usually have several schemata in Walenty, which leads to the average of 33 lexical entries per verb (even taking into account several schemata having

the same sub-schema, e.g., counting a singleton subject as one entry). Maximal number of lexical entries generated this way is 813 for the verb *dać* 'to give'. To make things worse, frequent verbs have more complicated valency than less frequent ones. If we take into the account frequencies of verbs we arrive at the average number of POLFIE style lexical entries equal 76 (counted on the Składnica corpus).

The solution used in POLFIE seems to be motivated by the limitations of LFG (or its implementation XLE), namely by the need to pass the valency information through lexicon entries. We have decided to take a different route. In DCG we have the advantage of being able to program arbitrary conditions, as if extending the formalism for the needs of a particular grammar. In particular, we can manipulate complex valency information during parsing.

We have decided to represent valency information in a form close to the source form of Walenty: a complete list of schemata for the given verb is passed to the parser (both reflexive and non-reflexive readings). Each schema is a list of syntactic positions. Each position is a list of argument specifications.

3.2 Filling syntactic positions

When the parser builds a node for a finite sentence it collects dependents for the given verb or rather for a verbal phrase with this verb as the centre. (We use the finite sentence as an example here, but the same type of processing occurs at all places when arguments are expected by some entity, be it a verb, a noun, an adjective or an adverb). The algorithm maintains two lists: a list of already recognised arguments and a structure representing arguments that can still be added to the interpretation being constructed. The first list is initialised as empty, the second – with the complete valency entry for the verb.

When a new candidate for an argument is considered the following operations need to be performed:

1. Find the set of all schemata that contain positions that contain the type of the given argument.

2. From all of these schemata remove the position containing the argument in question. Note that positions are understood as alternatives: when one argument realising a position is recognised, the whole position is removed as already realised. The result becomes the new list of not yet realised arguments.

3. Add the current argument to the list of already recognised arguments.

These steps are repeatedly applied to all arguments of the verb found in a given sentence.

length	1	2	3	4	5	6	7	8
no. of schemata	282	10701	29048	14419	2897	427	77	3

Table 1: Schema lengths in Walenty

3.3 Argument specifications

As said above, syntactic positions are sets (technically: lists) of argument specifications. These specifications again have some internal structure.

First of all to ease the processing we have decided to represent explicitly the information whether a given argument is obligatory (`obg`) or optional (`opt`).

In Walenty all arguments are optional with the following exceptions:

- All lexicalised (phraseological) arguments are obligatory.[3]
- An argument marked as `controller` is obligatory if its `controllee` is present.

In Świgra we treat the reflexive marker *się* as a special type of argument. This argument is obligatory in finite uses of verbs, but when a schema is derived for a gerund, the argument becomes optional. It is skipped completely when a schema for past participle is derived.

When the parsing algorithm finishes processing of arguments, the list of not yet used parts of schemata is checked against obligatory arguments. All schemata containing unrealised obligatory arguments are deleted from the list. The interpretation is accepted if the resulting list of schemata is nonempty, which means there was at least one schema whose all obligatory arguments were realised.

The second, most obvious, element of argument specification is its type, represented exactly as in the source dictionary.

The third part can contain additional information that further restricts the arguments. For example, a canonical subject is represented by the triple

```
opt/np(str)/subj(G,N,P)
```

where `G`, `N`, and `P` are Prolog variables unified by the algorithm with the values of gender, number, and person of the verb. When a given nominal phrase is to become a subject, its values of the respective categories are required to unify so that an agreement is maintained. A similar mechanism is used to enforce agreements between arguments resulting from the control relations described in Walenty.

3.4 Arguments coordinated within a position

To allow for unlike coordination rules were added to the grammar that allow required phrases `fw` to form coordinated structures. An example can be seen in Fig. 1, where required phrases `fw` for *pić* 'to drink' and *papierosa* 'a cigarette' get coordinated with the conjunction *i* 'and' and form a complex required phrase. The resulting required phrase has as its type a list of types of phrases that got coordinated. When matching such an argument against a syntactic position the algorithm checks whether all types in the list are allowed for the given position.

[3] In Świgra we do not yet use schemata containing lexicalised arguments, since for that the grammar itself will have to undergo some form of lexicalisation.

3.5 Example analysis

As an example let us consider the analysis of the following sentence:

(2) *Jan chce, żeby dać mu spokój.*
John want that give him peace

'John wants to be left alone.'

For brevity we list only a few of schemata for the verb CHCIEĆ and we skip control requirements and the obligatory/optional marker. The schemata in Świgra notation take the following form:

```
[ % schema 1
  [[sie], [np(dat)], [infp(_)]],
  % schema 2
  [[np(str)/subj(G,N,P)],
   [np(str), cp(żeby), infp(_), advp(misc)]],
  % schema 3
  [[np(str)/subj(G,N,P)],
   [np(gen), cp(żeby), ncp(gen,żeby)],
   [prepnp(od,gen)]]
]
```

When parsing example (2) the first argument encountered by the parser (working from the left to the right) is the nominal subject *Jan* of type `np(str)`. Since its morphological features agree with that of the verb we can accept this argument. This will result in filtering out schema 1, since it does not contain a subject. Then the subject position will be removed from schema 2 and 3 resulting in:

```
[% schema 2:
  [[np(str), cp(żeby), infp(_), advp(misc)]],
 % schema 3:
  [[np(gen), cp(żeby), ncp(gen,żeby)],
   [prepnp(od,gen)]]
]
```

The second argument is a clause , *żeby dać mu spokój* headed with the complementizer *żeby*. Its type `cp(żeby)` appears in both available schemata. After this step the list becomes:

```
[% schema 2:
  [],
 % schema 3:
  [[prepnp(od,gen)]]
]
```

To finish up we have to check whether any obligatory arguments remain unrealised, but that is not the case. The only obligatory argument was the reflexive marker `sie`. Both schema 2 and 3 allow to finish analysis at this stage.

It is worth noting that the recognised set of arguments can be an instance of schema 2 as well as schema 3. We do not differentiate between them and so only one parse tree gets generated.

4 Some experimental results

Świgra with Walenty dictionary and the adapted grammar was used to parse anew the whole Składnica corpus (20,000 sentences). This version was able to accept 14,103 sentences (70.5%), while the version with the old dictionary accepted 13,194 (66%). Unfortunately, these newly accepted sentences have not yet been validated by the annotators, so we cannot claim that all new structures are correct.

We have checked the structures generated using Walenty against 10,673 already accepted trees of Składnica. The tree previously accepted by the annotators was found among new parses in 10193 cases (95.5%). For the remaining 480 sentences (4.5%) the parser using Walenty did not produce a compatible tree (in 255 cases (2.4%) the new parse forest was empty). These cases will have to be studied carefully, since they show several problems including errors both in Składnica and in Walenty. For some verbs the two dictionaries differ in opinion whether a given dependent should be considered a complement or an adjunct, so these cases will require further discussion.

Since unlike coordination is one of the more advertised features of Walenty, we have also made a preliminary attempt to estimate the frequency of arguments being coordinated in that manner. The rules for coordination within positions were used in 141 sentences of 14103 sentences that were accepted by the parser. We have checked manually all these sentences and found that only 4 are real examples of this type of coordination, which amounts to 0.03% of sentences. This result can be biased by sentences rejected by the parser, but it seems to be in contrast with the claim of Patejuk & Przepiórkowski (2014) that "such coordination of unlike categories is relatively common in Polish."

5 Conclusions

Parsing Polish is to much extent valency driven. Valency schemata for Polish are numerous and complicated. Polish has free word order allowing to shuffle the schemata arbitrarily. Moreover, most of arguments of a verb are optional. These facts pose specific problems in parsing.

In the paper we have shown that with respect to these problems the DCG formalism provides tools leading to a more effective solution than LFG. One problem of this solution is that it has a "procedural" and not purely "constraint based" flavour. We think of it in terms of "when the parser recognises a candidate argument…",

"a position is removed from the schema…", etc. It seems that to express a similar solution in a constraint based formalism like LFG of HPSG some extensions would be needed in these formalisms.

We hope that this humble contribution will provide some food for thought on desirable features of a formalism well suited for parsing languages typologically similar to Polish.

References

Hajnicz, E., Nitoń, B., Patejuk, A., Przepiórkowski, A., & Woliński, M. (2015). Internetowy słownik walencyjny języka polskiego oparty na danych korpusowych. *Prace Filologiczne*, *LXV*, (to appear).

Patejuk, A. (2015). *Unlike coordination in Polish: an LFG account*. Ph.D. dissertation, Institute of Polish Language, Polish Academy of Sciences, Kraków.

Patejuk, A., & Przepiórkowski, A. (2014). Synergistic development of grammatical resources: a valence dictionary, an LFG grammar, and an LFG structure bank for Polish. In V. Henrich, E. Hinrichs, D. de Kok, P. Osenova, & A. Przepiórkowski (Eds.) *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT 13)*, (pp. 113–126). Tübingen, Germany: Department of Linguistics (SfS), University of Tübingen.
URL http://tlt13.sfs.uni-tuebingen.de/tlt13-proceedings.pdf

Pereira, F., & Warren, D. H. D. (1980). Definite clause grammars for language analysis–a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, *13*, 231–278.

Przepiórkowski, A., Bańko, M., Górski, R. L., & Lewandowska-Tomaszczyk, B. (Eds.) (2011). *Narodowy Korpus Języka Polskiego*. Warsaw: Wydawnictwo Naukowe PWN.

Przepiórkowski, A., Hajnicz, E., Patejuk, A., & 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)*, (pp. 83–91). Dublin, Ireland: Association for Computational Linguistics and Dublin City University.
URL http://www.aclweb.org/anthology/siglex.html#2014_0

Przepiórkowski, A., Hajnicz, E., Patejuk, A., Woliński, M., Skwarski, F., & 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, & S. Piperidis (Eds.) *Proceedings of the Ninth*

International Conference on Language Resources and Evaluation, LREC 2014, (pp. 2785–2792). Reykjavík, Iceland: ELRA.
URL http://www.lrec-conf.org/proceedings/lrec2014/index.html

Przepiórkowski, A., Skwarski, F., Hajnicz, E., Patejuk, A., Świdziński, M., & Woliński, M. (2014c). Modelowanie własności składniowych czasowników w nowym słowniku walencyjnym języka polskiego. *Polonica*, *XXXIII*, 159–178.

Świdziński, M. (1992). *Gramatyka formalna języka polskiego*. Rozprawy Uniwersytetu Warszawskiego. Warszawa: Wydawnictwa Uniwersytetu Warszawskiego.

Świdziński, M., & Woliński, M. (2010). Towards a bank of constituent parse trees for Polish. In P. Sojka (Ed.) *Text, Speech and Dialogue, 13th International Conference, TSD 2010, Brno, September 2010, Proceedings*, vol. 6231 of *LNAI*, (pp. 197–204). Heidelberg: Springer.

Woliński, M. (2004). *Komputerowa weryfikacja gramatyki Świdzińskiego*. Ph.D. thesis, Instytut Podstaw Informatyki PAN, Warszawa.

Woliński, M., Głowińska, K., & Świdziński, M. (2011). A preliminary version of Składnica—a treebank of Polish. In Z. Vetulani (Ed.) *Proceedings of the 5th Language & Technology Conference*, (pp. 299–303). Poznań.