Noname manuscript No.
(will be inserted by the editor)

From Lexical Functional Grammar to Enhanced Universal Dependencies

The UD-LFG Treebank of Polish

Adam Przepiórkowski ·
Agnieszka Patejuk

the date of receipt and acceptance should be inserted later

Abstract The paper describes the conversion of an LFG treebank[1] of Polish into enhanced Universal Dependencies, and – more generally – identifies the kinds of information lost in translation from LFG to UD. The paper also presents the resulting UD treebank of Polish and compares it to the previous UD treebank of Polish.

Keywords LFG · UD · Polish · coordination · heads

1 Introduction

Universal Dependencies (UD; Nivre et al. 2016) has recently become a *de facto* standard as a dependency representation used in Natural Language Processing (NLP). As most syntactic processing in NLP involves dependency structures, it is safe to say that it is becoming a standard for syntactic processing at large. As of early September 2018, there are 132 treebanks for 74 languages publicly available at `http://universaldependencies.org/`,[2] with 15 upcoming treebanks for a further 13 languages. New UD treebanks are often the result of converting corpora adhering to other annotation schemes – not only dependency-based, but also constituency-based.

A. Przepiórkowski
Institute of Philosophy, University of Warsaw, *and*
Institute of Computer Science, Polish Academy of Sciences
E-mail: adamp@ipipan.waw.pl

A. Patejuk
Institute of Computer Science, Polish Academy of Sciences, *and*
Faculty of Linguistics, Philology and Phonetics, University of Oxford
E-mail: aep@ipipan.waw.pl

[1] The term *treebank* is understood here as any syntactically annotated corpus, even if – as in the case of LFG annotations – not all syntactic structures assigned to sentences are trees.

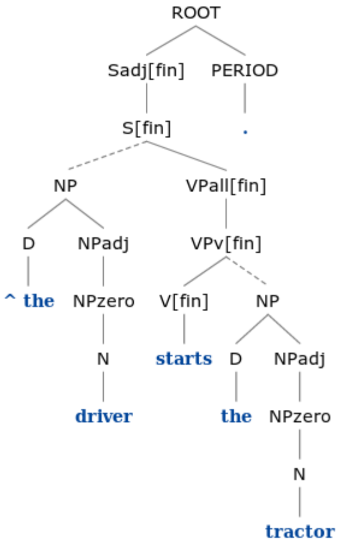
[2] All URLs in this paper were last accessed on 7 September 2018.

October 2018 version accepted to *Language Resources and Evaluation* – should appear in 2019.

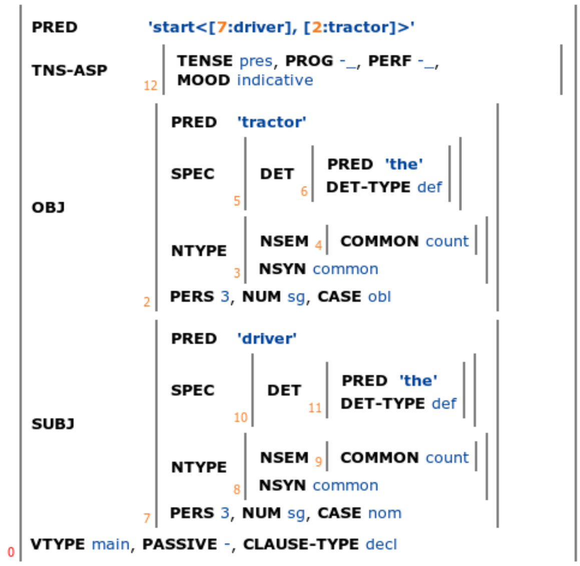
Lexical Functional Grammar (LFG; Bresnan 1982, Dalrymple 2001) is a linguistic theory which assumes two syntactic levels of representation (in addition to other, non-syntactic levels): constituency structure (c-structure) and functional structure (f-structure). In the case of the English sentence (1), from the multilingual LFG test-suite ParGramBank (Sulger et al. 2013), the c-structure is given in (2) and the f-structure – in (3):[3]

(1) The driver starts the tractor.

(2)



(3)



The first aim of this paper is to describe a procedure of converting such LFG structures to dependency representations following the UD standard, specifically, its enhanced version 2 (see §§2–3). Conversion of LFG structures to dependency structures is not a new task (cf. e.g. Øvrelid et al. 2009, Çetinoğlu et al. 2010 and, more recently, Meurer 2017), but – with the notable exception of Meurer 2017 – previous attempts are only mentioned or very roughly outlined in the literature.[4] Moreover, previous work has been limited to *dependency trees* as the output format. As is well known, simple dependency

[3] LFG constituency and functional structures given in this paper are visualisations of such structures produced by the INESS system (`http://clarino.uib.no/iness/`; Rosén et al. 2012), which hosts ParGramBank and the LFG treebank of Polish (called `pol-lfg` there), among other treebanks. While the complete f-structure is given in (3), subsequent f-structures are much simplified (limited to PRED values and relations between them).

[4] There is also some work devoted to conversion *to* LFG structures: from pure constituency treebanks (e.g. van Genabith et al. 1999 and later work referenced in Forst 2003), from constituency treebanks containing some dependency information (Forst 2003), and from pure dependency treebanks (Haug 2012).

trees cannot straightforwardly represent many kinds of linguistic information, so the conversion from representations such as those assumed in LFG invariably resulted in a considerable loss of information.

The current version 2 of Universal Dependencies assumes, apart from basic dependency trees, also *enhanced dependency structures*, which make it possible to represent phenomena beyond the descriptive power of simple trees. The second aim of this paper is to examine to what extent rich information available in LFG structures is or may in principle be preserved in such enhanced UD representations (see §4).

The empirical basis for the conversion is a manually disambiguated LFG parsebank of Polish (Patejuk and Przepiórkowski 2014b, 2018) consisting of over 17,000 sentences (almost 131,000 tokens). Since this is a parsebank, it only contains analyses successfully provided by the LFG parser of Polish (Patejuk and Przepiórkowski 2012, 2015b) and selected by human annotators as correct. While this constrains the number and kinds of constructions present in the corpus, the underlying LFG grammar of Polish is currently one of the largest implemented LFG grammars, and it includes a comprehensive analysis of various kinds of coordination and its interaction with other phenomena, so there is no shortage of sentences which pose potential difficulties for the conversion. The third aim of this paper is to present the resulting treebank of Polish, $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$(see §5), and to compare it with the previous UD treebank of Polish, $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$(see §6). Hence, the current paper is meant to become the standard reference for $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$.

2 From LFG to LFG-like dependencies

There is some disagreement about which syntactic level of representation – c-structure or f-structure – is the most natural basis for constructing dependency representations. While f-structure seems to be a natural candidate, Meurer 2017 sketches a conversion procedure based mainly on c-structure and consisting in stepwise transformations of the constituency tree into a dependency tree.

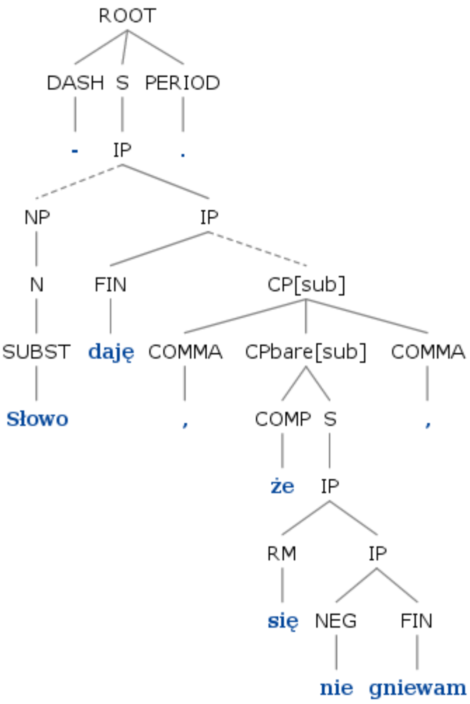
The approach presented here follows the more standard observation that f-structures provide a good basis for dependency relations. For example, apart from much morphosyntactic information (e.g., that the sentence is in indicative mood and present tense, and is not a progressive or perfective form, or that the subject is in the nominative case, 3rd person singular), which will be omitted in subsequent f-structures, the f-structure in (3) contains information about two dependency relations encoding grammatical functions: the subject dependency (from *starts* to *the driver*) and the (direct) object dependency (from *starts* to *the tractor*).

Of course, c-structures cannot be completely ignored, as only they contain the actual tokens in the sentence (f-structure PRED values usually use lemmata as functors, e.g. START instead of *starts*) and information about their linear order. We show that – apart from f-structures – information encoded in terminal and pre-terminal nodes of the constituency tree, together with the standard correspondence between c-structure preterminals and f-structure components,[5] is sufficient to perform the conversion, i.e., that the actual constituency information may be completely ignored.

Conversion is performed in two stages: from LFG syntactic structures to initial dependency structures which closely mirror LFG representations, and from such initial dependency structures to final enhanced UD representations. The sole difficulty of the first stage stems from the fact that preterminals of multiple tokens often map to the same functional structure, as illustrated with example (4), whose c-structure is given in (5) and f-structure – in (6).

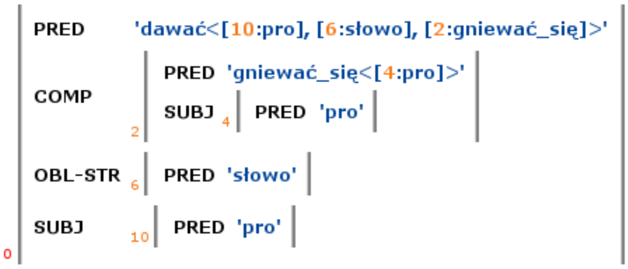
(4) - Słowo daję, że się nie gniewam.
word.ACC give.1SG that RM NEG be_angry.1SG
'I give you my word that I am not angry.'

(5)



[5] This correspondence, often called ϕ, is a function from non-terminal nodes in c-structure to particular substructures in f-structure. For example, in the case of (2)–(3), the leftmost nodes NP, NPadj, NPzero, N and D in (2) all map to the substructure with index 7 (i.e., the value of SUBJ) in (3), the rightmost nodes NP, NPadj, NPzero, N and D all map to the substructure with index 2 (i.e., the value of OBJ), and all the other nonterminals, including ROOT, S[fin] and V[fin] – to the whole f-structure with index 0. In order to avoid clutter, such correspondences will not be explicitly shown in figures below, but they will be pointed out in the text, where needed.

(6)



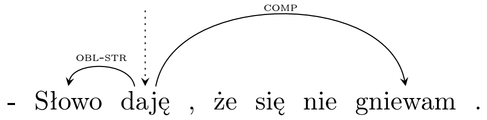
```
   PRED       'dawać<[10:pro], [6:słowo], [2:gniewać_się]>'
              PRED 'gniewać_się<[4:pro]>'
   COMP       SUBJ 4 [ PRED 'pro' ]
            2
   OBL-STR  6 [ PRED 'słowo' ]
   SUBJ    10 [ PRED 'pro' ]
0
```

Of the five feature (sub)structures in (6), two do not correspond to any token (substructures with indices 4 and 10; they represent *pro*-dropped subjects), so the ten preterminals in (5) – corresponding to the nine tokens in the sentence (including punctuation)[6] – map to just the remaining three such f-structures (those with indices 0, 2 and 6).

2.1 True heads

There are two problems to be solved at this stage. The first problem amounts to deciding which of the co-heads – tokens mapping to the same f-structure – is the true head. Once this is decided, grammatical functions relating f-structures, for example, the COMP function relating the matrix f-structure with index 0 and the embedded f-structure with index 2, may be directly translated into dependencies between the true heads corresponding to these f-structures. This provides the backbone for the LFG-like dependency structure:

(7)



The true head is chosen mainly on the basis of part-of-speech information: in this case, the verb *daję* 'give' wins the competition with the two (sentence-initial and sentence-final) punctuation marks, and the verb *gniewam* 'be angry' wins with the negative marker (NEG) *nie*, the so-called reflexive marker (RM) *się* (here, an inherent part of the verb form, without anaphoric meaning), the complementiser *że* 'that' and the comma; hence, the COMP dependency between *daję* and *gniewam* in (7). Similarly, since *słowo* 'word' has no co-heads, it is trivially the true head mapping to the f-structure with index 6 in (6), so the OBL-STR grammatical function relating f-structures 0 and 6 in (6) is translated into the OBL-STR dependency between *daję* and *słowo* in (7).[7]

[6] The final comma in (5), marking the end of the subordinate clause, is added by the tokeniser at the stage of LFG parsing.

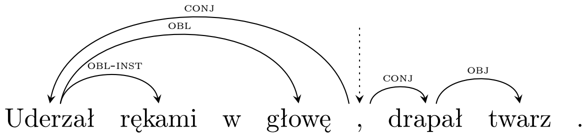
[7] In the Polish LFG treebank, the values of OBL-STR are those subcategorised obliques (non-subjects and non-objects) which are in the structural case (Patejuk and Przepiórkowski 2014a), that is, in the accusative (as in (4)) or – in the presence of negation – in the genitive.

In the case of coordination, the conjunction is selected as the head, while set membership is translated as CONJ dependency. This is illustrated with example (8), whose c- and f-structures are given in (11)–(12).[8] As this example involves asyndetic coordination, the comma acts as the conjunction, so the two CONJ dependencies are from the comma to the two finite verbs which head the two conjuncts, i.e., to *uderzał* 'hit, pounded' and to *drapał* 'scratched' – see the backbone of the LFG-like dependency structure in (9).

(8)	Uderzał	rękami	w głowę,	drapał	twarz.
	hit.3SG.M	hands.INST	in head.ACC	scratched.3SG.M	face.ACC

'He pounded his head with his fists, scratched his face.'

(9)



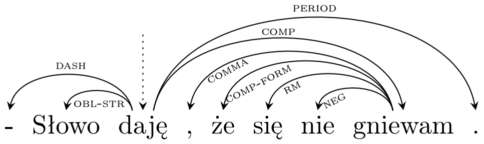
Note that the non-semantic preposition *w* 'in' does not introduce its own PRED value in (12) – it is a co-head of the noun *głowę* 'head'. Since the noun is selected as the true head in this case, it is the target of the OBL dependency from *uderzał* 'hit, pounded'.

2.2 Dependencies to co-heads

The second problem consists in deciding on dependency labels from true heads to their co-heads, e.g. from *gniewam* 'be angry' to *nie* (negative marker), *się* ('reflexive marker', devoid of any reflexive meaning in this case), *że* (complementiser) and the comma in (7), or from *głowę* 'head' to *w* 'in' in (9). The solution adopted here is trivial: dependencies to co-heads are labelled with the names of the preterminals of these co-heads in c-structure.

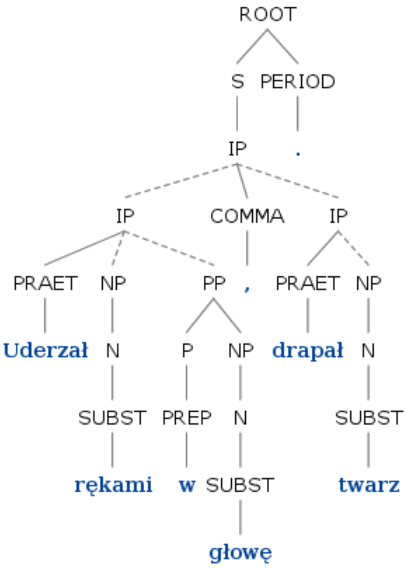
The result of this first stage of conversion is given in (10) in the case of the first example, (4), and in (13) – in the case of the second example, (8). Most of the added dependencies – DASH, PERIOD, COMMA, RM and NEG in (10), and PREP and PERIOD in (13) – directly correspond to the preterminals of the relevant tokens in c-structures (5) and (11). The only exception is made for the complementiser, whose preterminal is COMP, since it accidentally bears the name of a standard LFG grammatical function (used to mark non-subject clausal arguments). For this reason, the corresponding dependency relation in (10) is called COMP-FORM.

(10)

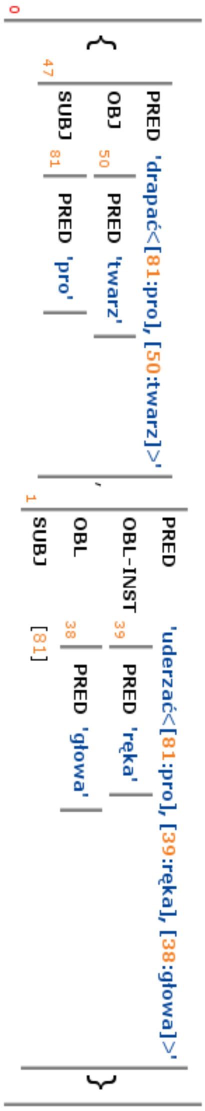


[8] The INESS system which produces such visualisations (see fn. 3) does not necessarily display set elements in their linear order. Apart from (12), this also affects f-structures (31) and (44).

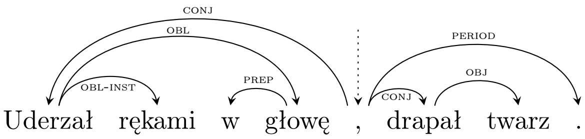
(11)



(12)



(13)



2.3 Dependency graph and dependency tree

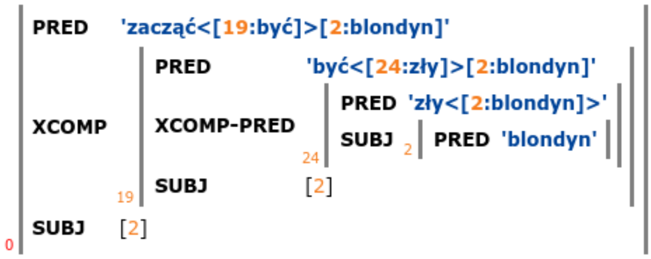
While (10) and (13) are dependency *trees*, the result of the first stage of conversion may be a more complex dependency *graph* in which a number of heads share dependents. There are three main phenomena which give rise to such more complex representations: broadly understood control (including raising), predicative dependents and coordination with shared dependents. The following example illustrates the first two (see (25) below for the last one):

(14)	Blondyn	zaczął	być	zły.
	blond.NOM.SG.M	began.3SG.M	be.INF	angry.NOM.SG.M

‘The blond guy started to be angry.’

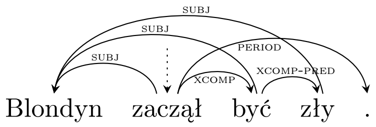
As is made clear in the f-structure (15), *blondyn* ‘blond guy’ is the subject of the raising verb *zaczął* ‘started’, but it is also the understood subject of the copula *być* ‘be’ (also analysed here as a raising verb) and of the predicative adjective *zły* ‘angry’.

(15)



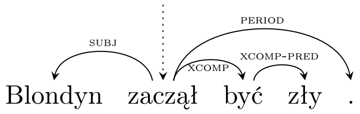
These dependencies are truthfully reflected in the dependency graph (16):

(16)



Such dependency graphs constitute input to the second stage of conversion, described in the following section. The result of the second stage is the final UD representation, which consists of two structures: an enhanced dependency graph, where dependents may have multiple heads, and a basic dependency tree, where each dependent has exactly one head. For this reason, dependency graphs resulting from the first conversion stage are also simplified to pure dependency trees, as in (17), in the case of the example at hand:

(17)

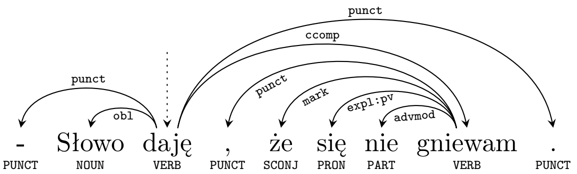


The two structures – the full dependency graph and the basic dependency tree – are subsequently processed in parallel, in the way described in the following section, resulting in the two UD representations of the sentence.

3 From LFG-like dependencies to enhanced UD

In the simplest – but rare – case, in order to arrive at the final UD representation, it is sufficient to rename LFG dependency labels to UD labels, as shown in (18), to be compared to (10) (on page 6).[9]

(18)


punct
ccomp
punct
obl
punct
mark
expl:pv
advmod
- Słowo daję , że się nie gniewam .
PUNCT NOUN VERB PUNCT SCONJ PRON PART VERB PUNCT

This final UD representation differs from the LFG-like dependency structure in (10) only in the names of the labels: various LFG labels for punctuation dependents (DASH, COMMA, PERIOD) are translated into the UD label `punct`, labels marking oblique dependents (here, OBL-STR) – into `obl`, COMP – into `ccomp`, COMP-FORM – into `mark`, RM – into `expl:pv`,[10] NEG – into the general dependency label used to mark various adverbial modifiers, `advmod`.

3.1 Reversing dependencies

However, in the usual case, initial dependency structures must also be rearranged, for two main reasons. The first reason is that UD adopts the principle of the primacy of content words – rather than function words – as heads. This means that, unlike in LFG representations, prepositional phrases are headed by nouns (even in the case of semantic prepositions, which contribute a PRED value), numeral phrases are headed by nouns (even though, for Polish, there are good arguments to the contrary), and auxiliaries and copulas are always dependents, rather than heads. This is not only a matter of reversing single dependencies: all dependencies originally targeting the functional head must now target the content head, and all outgoing dependencies from the functional head must now originate in the content head.

This kind of restructuring is not needed in the case of the two examples introduced so far (but see the next subsection, §3.2, about restructuring coordination in one of them). The first one, (4) – with the LFG-like dependency representation in (10) and the UD representation in (18) – has already been discussed, and the second, (8) – whose LFG-like dependency representation is

[9] UD dependency labels are distinguished from LFG-like dependency labels typographically; the latter are written in small capitals, e.g. COMP, and the former in monospace, e.g. `ccomp`. Additionally, UD representations contain information about the coarse part of speech (UPOS, in the CoNLL-U representation used in UD) of each token. The basic dependency tree is displayed above the tokens, and the enhanced dependency representation is shown below the tokens. In (18), since the enhanced dependency representation is the same as the basic dependency tree, only the latter is displayed.

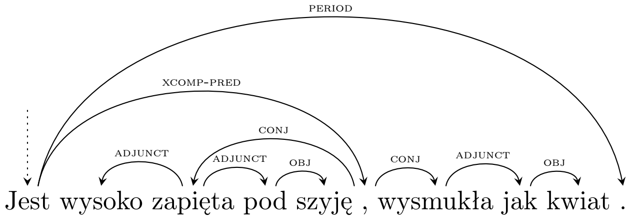
[10] `expl:pv` is a relation commonly used in UD to mark those occurrences of the so-called reflexive marker in Slavic which are inherent parts of verbs.

given in (13) (and the UD representation – in (24) in the next subsection) – involves a non-semantic preposition, represented as a dependent of the noun, so the dependency between this preposition, *w* 'in', and the noun, *głowę* 'head', already satisfies the UD principle of the primacy of content words. Consider, however, example (19), with c- and f-structures in (22)–(23).

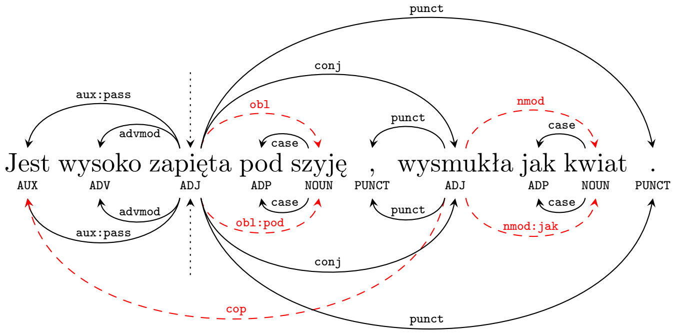
(19) Jest wysoko zapięta pod szyję, wysmukła
is.3SG highly buttoned_up.NOM.SG.F under neck.ACC lean.NOM.SG.F
jak kwiat.
like flower.NOM.SG.M
'She is buttoned up high to the neck, lean like a flower.'

There are two prepositional phrases in this example, both involving semantic prepositions: *pod szyję* 'up to the neck', literally: 'under neck', and *jak kwiat* 'like (a) flower'. Unlike the non-semantic preposition in the previous example, (8), these two semantic prepositions introduce their own PRED values, i.e., they project their own f-structures (with indices 32 and 56 in (23)), containing functional representations of the embedded noun phrases as values of OBJ. Hence, the initial LFG-like dependency representation contains OBJ dependencies with prepositions as heads and corresponding nouns as dependents – see (20). At the second stage of conversion, these dependencies have to be reversed (such dependencies from nouns to prepositions are labelled `case` in UD), and the dependencies targeting the whole prepositional phrase – i.e., the two ADJUNCT edges targeting the prepositions – have to be renamed and rearranged so that they target the new heads of prepositional phrases, i.e., the two nouns; see the final UD representation in (21), to be discussed shortly.

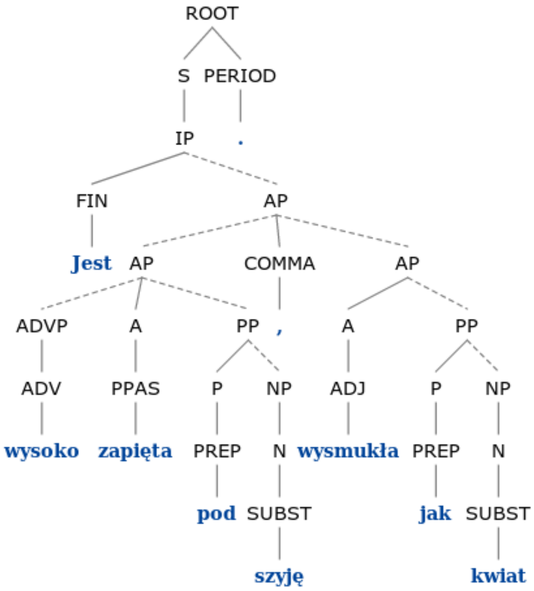
(20)



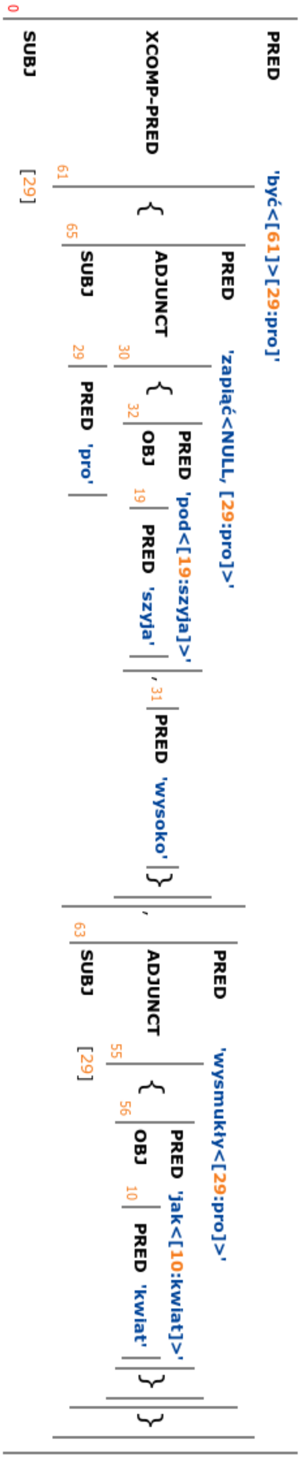
(21)


punct
conj
aux:pass
obl
advmod
case
punct
nmod
case
Jest wysoko zapięta pod szyję , wysmukła jak kwiat .
AUX ADV ADJ ADP NOUN PUNCT ADJ ADP NOUN PUNCT
advmod
case
punct
case
aux:pass
obl:pod
nmod:jak
conj
cop
punct

(22)


ROOT
S
PERIOD
IP
.
FIN
AP
Jest
AP
COMMA
AP
ADVP
A
PP
,
A
PP
ADV
PPAS
P
NP
ADJ
P
NP
wysoko
zapięta
PREP
N
wysmukła
PREP
N
pod
SUBST
jak
SUBST
szyję
kwiat

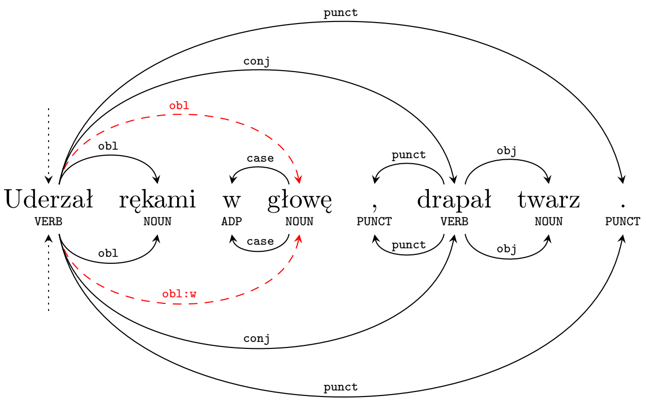
(23)


0
PRED 'być<[61]>[29:pro]'
XCOMP-PRED
61
PRED 'zapiąć<NULL, [29:pro]>'
ADJUNCT
30
32
PRED 'pod<[19:szyja]>'
OBJ
19
PRED 'szyja'
31
PRED 'wysoko'
65
SUBJ
29
PRED 'pro'
63
PRED 'wysmukły<[29:pro]>'
ADJUNCT
55
56
PRED 'jak<[10:kwiat]>'
OBJ
10
PRED 'kwiat'
SUBJ [29]
SUBJ [29]

3.2 Rearranging dependencies in coordination

The second reason for rearranging dependency structures is that UD adopts a representation of coordination in which it is the first conjunct – not the conjunction – that is the head. All other conjuncts are direct dependents of this head and the conjunction is a dependent of the conjunct to its right.[11] Let us first exemplify this structural change with the above example (8), whose initial LFG-like dependency representation is given in (13) (on page 7), and the final UD representation – in (24):

(24)



Here the enhanced UD representation, drawn below the sentence, differs from the basic UD representation, drawn above the sentence, although the difference is trivial: in the enhanced representation, the dependency label to the prepositional phrase *w głowę* 'in (the) head', `obl:w`, is subtyped with the preposition *w*. (Any dependencies which are not identical in the two UD representations are shown in red with dashed edges.)

As the comparison of (13) and (24) illustrates, rearranging dependencies in coordination may result in a change of the root of the sentence: in the LFG-like representation (13), the comma, which acts as a conjunction, is the root, but in the final UD representation (24), the head of the first conjunct, i.e. *uderzał* 'hit, pounded', is the root. As the final period in the sentence is analysed as a dependent of the root of the sentence, it must now be made the dependent of this verb, rather than of the comma conjunction. Further, the comma is now made a dependent of the (head of the) immediately following conjunct,

[11] So this representation differs from that of Mel'čuk 1988, where the first conjunct is the head but other conjuncts and the conjunction form a chain. While an asymmetrical analysis of coordination, favouring the first conjunct, may seem surprising at first, similar asymmetrical analyses are argued for also within some transformational (constituency-based) approaches, see e.g. Munn 1993: ch.2, Johannessen 1996, and Zhang 2009: ch.3 (and references therein).

i.e., of *drapał* ‘scratched’, which in turn is now made a dependent of the first conjunct, rather than of the comma conjunction.

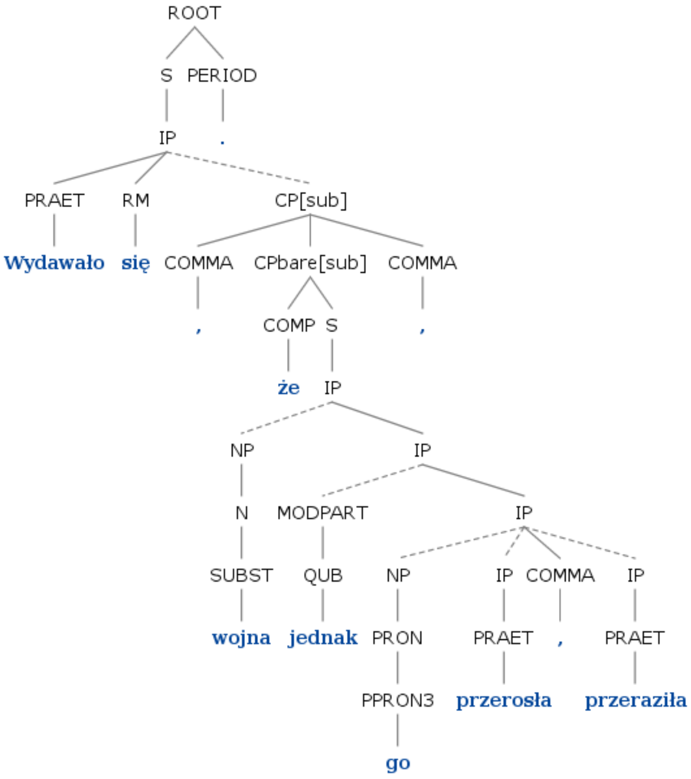
The complexity of this rearrangement is also seen in the case of the above example (19), whose initial and final dependency representations are given in (20)–(21). This example also involves asyndetic coordination, namely, of a passive participial phrase, *wysoko zapięta pod szyję* ‘buttoned up high to the neck’, and a predicative adjectival phrase, *wysmukła jak kwiat* ‘lean as a flower’. In the LFG-like dependency structure (20), the two conjuncts are dependents of the conjunction (here, the comma), and the whole coordinate structure is an XCOMP-PRED argument of *jest* ‘is’, which is the root of the whole sentence. On the other hand, in the basic UD representation (see the upper part of (21)), the coordinate structure is headed by the first conjunct, the passive participle, and the function word *jest* ‘is’ is its dependent (another case of the reversal of a dependency between a function word and a content word), so the passive participle is promoted to the status of the root of the sentence. This, and the fact that the relation between the passive participle *zapięta* ‘buttoned up’ and *jest* ‘is’ is `aux:pass`, means that the whole sentence is a passive construction according to this representation. But this is contingent on the linear order of the two conjuncts: were it opposite, the main relation would be that from the predicative adjective *wysmukła* ‘lean’ to *jest* ‘is’ – labelled as `cop` (copula) – so the whole sentence would in effect be analysed as a copular construction.[12] This dual nature of *jest* ‘is’ – as a predicative copula and as a passive auxiliary – is expressed in the enhanced dependencies (see the lower part of (21)), which include one more dependency: the `cop` edge from the adjective *wysmukła* ‘lean’ to *jest* ‘is’.

A much more robust example of dependent- and head-sharing is (25), whose c-structure and f-structure are given in (26)–(27).

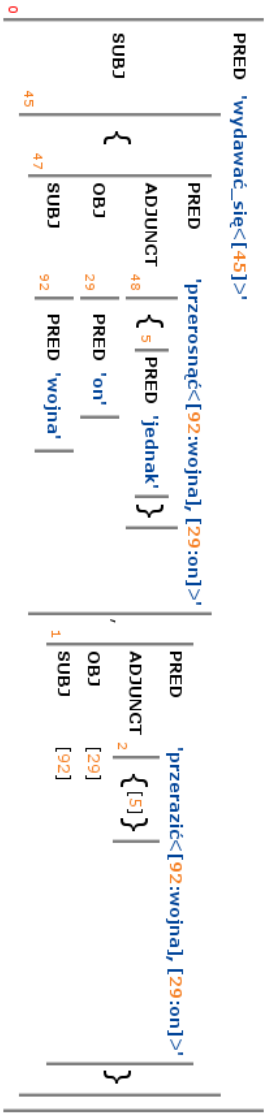
(25) Wydawało się, że wojna jednak go przerosła,
seemed.3SG.N RM that war.NOM.SG.F after all him.ACC overwhelmed.3SG.F
przeraziła.
scared.3SG.F
‘It seemed that, after all, the war overwhelmed and scared him.’

[12] Similar cases of determining the label on the basis of the order of conjuncts also occur elsewhere in UD, e.g., in the case of unlike nominal/clausal coordination in the subject position, where the label of the subject dependency is either `nsubj` or `csubj`, depending on the first conjunct. See Przepiórkowski and Patejuk 2018 for a proposal of a more transparent naming scheme.

(26)

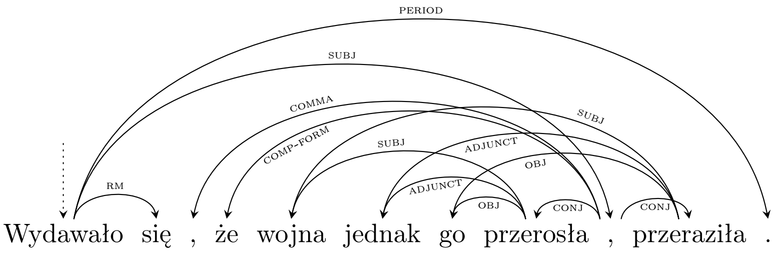


(27)

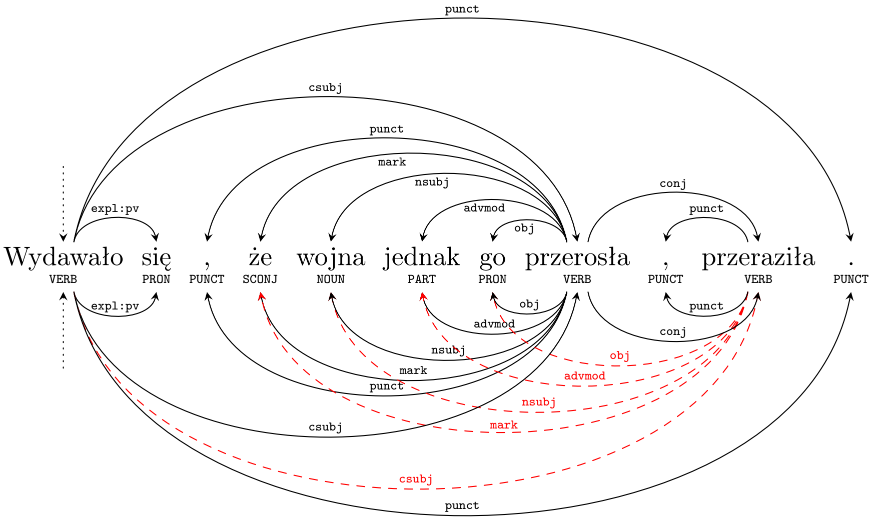


Here, the two asyndetically coordinated verbs, *przerosła* 'overwhelmed' and *przeraziła* 'scared', share their subject (*wojna* 'war'; substructure 92 in (27)), direct object (*go* 'him'; substructure 29 in (27)), and an adjunct (*jednak* 'after all'; substructure 5 in (27)). This structure-sharing is also reflected in the initial LFG-like dependency structure (28). Apart from these three explicitly shared dependents, there is another dependent of the coordinate structure as a whole, i.e., the non-semantic complementiser *że* ‘that’.[13] It is expressed in (28) as a COMP-FORM dependent of the coordinating comma.

(28)



(29)



In the UD representation, shown in (29), the head of the coordination is the first conjunct, *przerosła* ‘overwhelmed’, so – in the basic dependency tree (in the upper part of (29)) – these four shared dependents are represented as dependents of the first conjunct. Thus, this basic UD representation is ambiguous: each of the dependents of the first conjunct may be understood either as a dependent of the first conjunct alone, or as a dependent of the whole coordinate structure (i.e., of all conjuncts). This ambiguity is resolved in the enhanced representation (in the lower part of (29)): there, all four elements are also marked as dependents of the second conjunct, *przeraziła* ‘scared’.

[13] In the input LFG structures, a distinction is made between non-semantic complementisers, which do not project their own PRED, and semantic complementisers, which do; this distinction is analogous to that between non-semantic and semantic prepositions.

Additionally, as the whole coordinate structure is the subject of the main verb *wydawało się* 'it seemed', there is a `csubj` ('clausal subject') dependency from this verb to the first conjunct, matched by an analogous enhanced dependency to the second conjunct. In effect, the enhanced representation makes it possible to encode both dependent-sharing and head-sharing in coordination.

3.3 Partial conclusion

The first conversion stage, described in §2, is almost language-independent. For any language, relations between f-structures, such as SUBJ or XCOMP, can be translated into dependencies between the most important words – called true heads above – corresponding to these f-structures; other words corresponding to the same f-structure – co-heads – can be made dependents of such true heads, with dependency labels reflecting morphosyntactic categories of the co-heads. The only language-specific part at this stage is the heuristic selecting true heads among co-heads – this heuristic must be sensitive to morphosyntactic categories assumed for a given language.

The main idea of the second stage, described in this section, may also be applied to any language, but specific rules implementing the stepwise transformation of dependency structures depend not only on a given language, but also on particular analyses in the input LFG treebank. For example, as in the case of many other LFG treebanks, the Polish LFG treebank distinguishes not only between semantic and non-semantic prepositions, but also between semantic and non-semantic complementisers: the former, such as *ponieważ* 'because', introduce a semantic relation between clauses; the latter, such as *że* 'that', are meaningless markers of subordination. While non-semantic prepositions and complementisers are typically co-heads of, respectively, nouns and verbs, so they will be appropriately represented as dependents already after the first stage of conversion, semantic prepositions and complementisers are heads of the following nominal phrases and clauses in the LFG treebank, so at the second conversion stage they must be re-analysed as dependents by a rule that is specific to this treebank. Similarly, cardinal numeral phrases such as *pięć słów* 'five words' are commonly – but not universally – analysed in Polish formal linguistics as headed by numerals rather than by nouns, i.e., as true numeral phrases (Saloni and Świdziński 1985, Przepiórkowski 1999), and this analysis is adopted in the LFG treebank of Polish. Given that numerals are dependents of nouns in UD, this necessitates a rule reversing this dependency. In LFG treebanks for many other languages, and hypothetically also in a different LFG corpus of Polish, numerals may be represented as dependents of nouns, so such a rule would not be necessary.

Apart from reversing some dependencies, other language- or treebank-specific rules of the second stage of conversion concern the translation of LFG relations into UD labels. For example, in the $\text{UD}^{\text{PL}}_{\text{LFG}}$ treebank described here, direct objects are understood as any passivisable dependents (i.e., any dependents which become subjects in the passive voice), and indirect objects – as any arguments in the dative case. This leads to very simple translation rules, where UD dependency labels `obj` and `iobj` fully correspond to the LFG relations OBJ and OBJ-TH present in verbal f-structures.[14] As discussed in §6.3 below and – more fully – in Przepiórkowski and Patejuk 2018, some other UD treebanks, including the previous UD treebank of Polish, adopt a different approach to direct and indirect objects, which would call for very different – much more complex – translation rules from LFG relations.

To summarise, while the general approach presented above may be applied to any LFG treebank of any language, and the first stage of conversion only minimally depends on the specifics of the input LFG representation (namely, only on c-structure preterminals and their relative potential to be the true head), any implementation of the second stage is not only language-dependent, but also depends on particular analyses in the input LFG treebank and on the particular understanding of specific labels in the output UD treebank.[15]

4 Lost in translation

As is well known – and has already been pointed out above – dependency trees are less expressive than functional structures. One reason is that they do not make it possible to represent shared dependents, e.g. the fact that a dependent of a higher verb is at the same time the subject of the lower verb in control or raising constructions. For example, in (30), whose f-structure is given in (31), *Poczta* '(Polish) Postal Service', is the subject not only of the two conjoined finite verbs, *zmniejsza* 'reduces' and *powinna* 'should', but also of the controlled verbs *zacząć* 'start' and *przynosić* 'bring, make'. This is directly expressed in the f-structure (see the multiple occurrences of the substructure 156 in (31)), as well as in the LFG-like dependency representation (32), which is not a dependency tree, but a more complex graph. On the other hand, this information is lost in the basic UD tree in the upper part of (33).

(30)	Poczta	zmniejsza	swój	deficyt	i	już	w	1997	r.
	post.NOM.SG.F	reduces.3SG.F	self's	deficit	and	already	in	1997	year

	powinna	zacząć	przynosić	zyski.
	should.3SG.F	start.INF	bring.INF	profits.ACC

'Postal Service reduces its deficit and it should start to make profit already in 1997.'

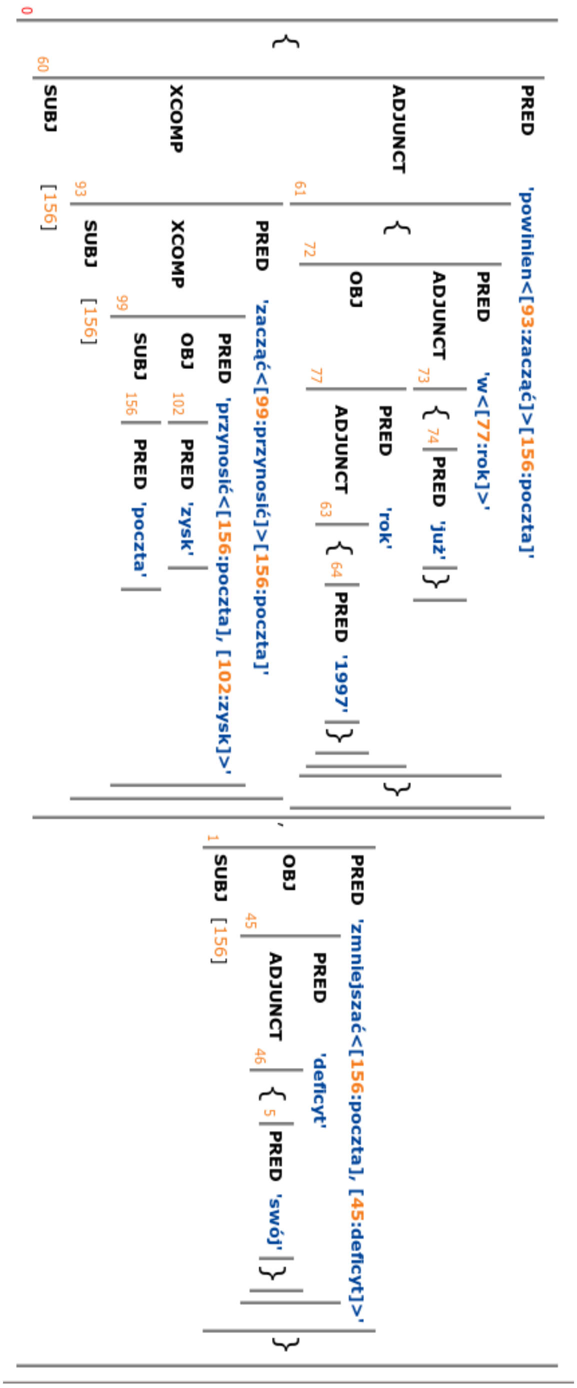
However, such information is easy to represent in the enhanced UD graph, which does not have to be a tree. Hence, in the case at hand, the information that four verbs share the same subject is not lost in the (enhanced) UD representation. The natural question is then, to what extent – if any – information is

[14] Recall that OBJ is also used to mark arguments of prepositions, hence the restriction to verbal f-structures.

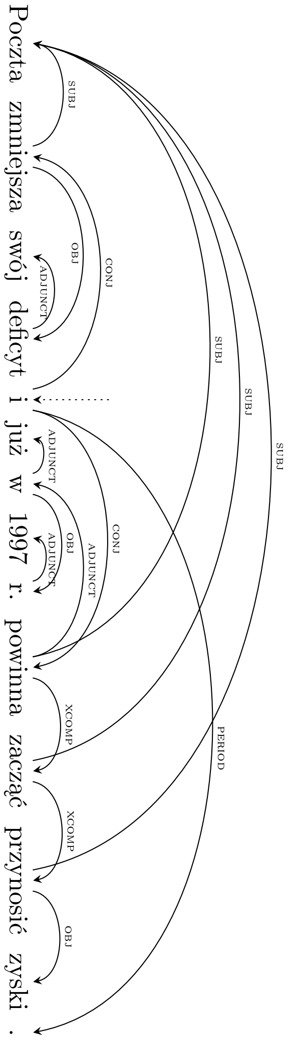
[15] As the second stage involves many such language-specific conversion steps, enumerating them here would be difficult for space reasons, but see Patejuk and Przepiórkowski 2018 for more detail.

lost in the translation from syntactic structures assumed in LFG to enhanced Universal Dependencies.

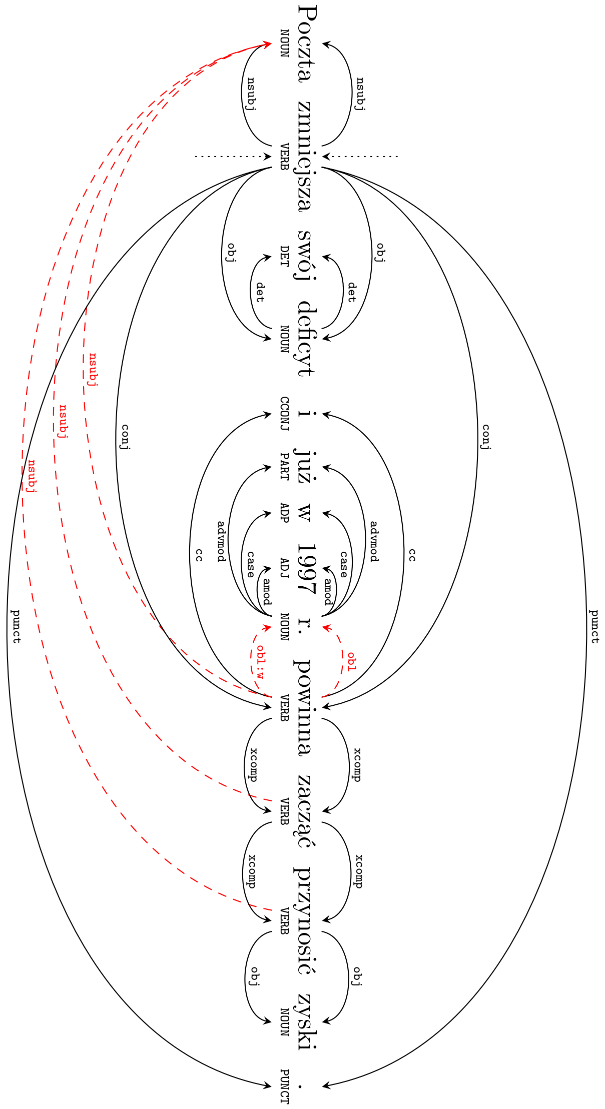
(31)



(32)


Poczta zmniejsza swój deficyt i już w 1997 r. powinna zacząć przynosić zyski .
SUBJ
OBJ
ADJUNCT
CONJ
SUBJ
SUBJ
SUBJ
ADJUNCT
OBJ
ADJUNCT
ADJUNCT
CONJ
XCOMP
XCOMP
OBJ
PERIOD

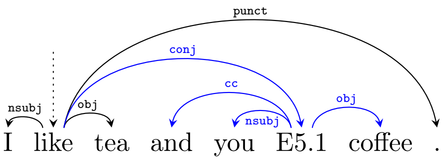
(33)


Poczta zmniejsza swój deficyt i już w 1997 r. powinna zacząć przynosić zyski .
NOUN VERB DET NOUN CCONJ PART ADP ADJ NOUN VERB VERB VERB NOUN PUNCT
nsubj
obj
det
conj
cc
advmod
case
amod
obl
obl:v
xcomp
xcomp
obj
punct
nsubj
nsubj
nsubj

4.1 Empty dependents not allowed

Clear loss of information results from the fact that UD does not make it possible to represent *pro*-dropped dependents. This is not a matter of a general ban on null nodes in dependency representations: enhanced UD makes it possible to represent elided *predicates*, as in the following example of the enhanced representation from the UD guidelines:[16]

(34)



Here, *E5.1* is an artificial token, added to the input sentence *in lieu* of the elided verb *like*. However, similar addition of tokens standing for *pro*-dropped *dependents* is currently prohibited, with the effect that information is lost in the conversion of some of the examples given above.

Consider again example (8) (on page 6) and its f-structure in (12) (on page 7). Polish is a rampantly *pro*-drop language, and in this sentence the *pro*-dropped subject is shared between the two finite verbs (see the substructure with index 81 in (12)). That is, the same person is understood to have done the pounding and the scratching. In contrast, the UD representation (24) (on page 12) misses this important information – it is underspecified as to whether the same person performed the two actions. A similar problem occurs in the case of example (19) (on page 10): while the f-structure (23) makes it clear that the passive participle and the predicative adjective share their subject (see the substructure 29 there), this information is missing in the UD representation in (21).

A related problem is that, in the case of the *pro*-drop of the controller, information is lost about the reference of the subject of the controlled verb. In the absence of *pro*-drop, this information is given explicitly in the enhanced representation; for example, in the UD representation in (33), the controlled verbs are those with the incoming `xcomp` dependency – i.e., *zacząć* 'start' and *przynosić* 'bring, make' – and their subjects are marked by the `nsubj` enhanced dependencies to *Poczta* 'Postal Service'. Consider, however, example (35), involving the control verb *kazał* 'ordered'.

(35)

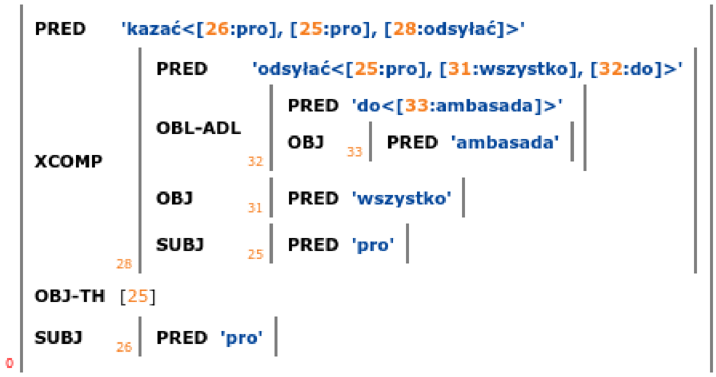
Kazał	wszystko	odsyłać	do	ambasady.
ordered.3SG.M	all.ACC	send_back.INF	to	embassy

'He ordered to send everything back to the embassy.'

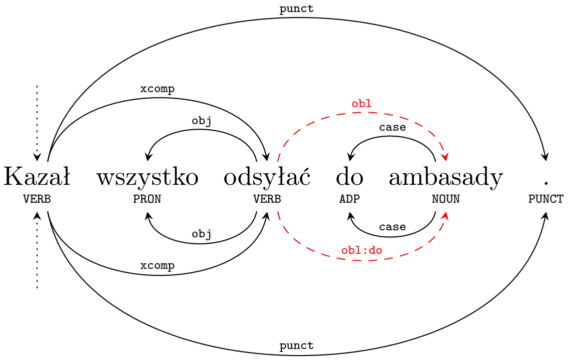
Two arguments of this verb are *pro*-dropped: the subject and the dative argument which controls the subject of the infinitival *odsyłać* 'send back'. This information is explicitly represented in f-structure (36). In particular, the SUBJect of the controlled verb, i.e., the substructure with index 25, is the same as the dative argument of the main verb, i.e., as the value of the OBJ-TH attribute there. Unfortunately, there is currently no way to represent this information in the UD structure – see (37).

[16] `http://universaldependencies.org/u/overview/enhanced-syntax.html#ellipsis`; the dependencies in blue are present only in the enhanced representation.

(36)



(37)



Another problem stemming from the lack of any representation of *pro*-dropped dependents concerns the representation of non-core (not subcategorised, not required) secondary predicates, e.g. *pierwszy* 'first' in (38) and *osłupiały* 'transfixed, shocked' in (39):[17]

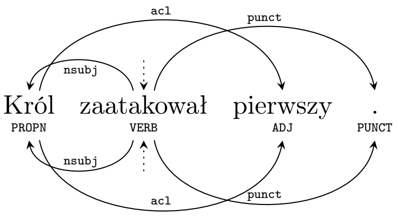
(38) Król zaatakował pierwszy.
king.NOM.SG.M attacked.3SG.M first.NOM.SG.M
'The king attacked (as) first.'

(39) Przez chwilę stał osłupiały.
for while stood.3SG.M transfixed.NOM.SG.M
'He stood transfixed for a while.'

Such non-core secondary predicates are `acl` dependents of the nouns they predicate of, as shown in (40).

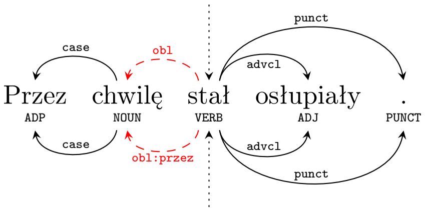
[17] An analogous problem occurs in the case of (subcategorised, required) predicative complements.

(40)



However, such an overt target of predication is missing in (39), in which case the secondary predicate should be an `advcl` dependent of the verb that governs the *pro*-dropped argument:

(41)



This not only results in rather different representations of the same phenomenon, but also representations such as (41) are in the general case underspecified as to which of the potentially *pro*-dropped dependents of the verb the predicate refers to.[18]

While the prohibition on explicit representation of *pro*-dropped dependents is probably the most important source of information loss in the conversion procedure described above, we do not see it as a fundamental problem with UD representation: once this arbitrary prohibition is lifted, the problems described in this subsection should disappear.

4.2 Multiple dependencies between same tokens not allowed

A statistically insignificant problem, but one that did occur in the conversion process, is that it is prohibited at the moment, even in enhanced dependencies, to have two different edges from token A to token B. The need for such a representation arises in those – admittedly very rare – cases where the multifunctional 'reflexive' marker *się* plays two roles at the same time (Patejuk and Przepiórkowski 2015a), e.g., being a marker of an 'inherently reflexive' verb (`expl:pv`, as in (18) and (29)), and being a part of an impersonal construction (`expl:impers`). A treebank example illustrating this problem is (42): the first *się*, in *uczestniczyło się* 'one participated', is purely impersonal, while the second *się*, in *modliło się* 'one prayed', is impersonal and also an inherent part of the verb MODLIĆ SIĘ 'pray', so it should bear two relations to *modliło*: `expl:impers` and `expl:pv`.

[18] On the other hand, in the case of morphologically rich languages such as Polish, the case value of the secondary predicate should make this clear in most – but not all – instances.

(42) W Laskach w liturgii uczestniczyło się przez cały dzień i
in Laski in liturgy participated.3SG.N RM.IMPS for whole day and
modliło się wszędzie.
prayed.3SG.N RM.INH.IMPS everywhere
'In Laski, one would take part in the liturgy for the whole day and one would pray everywhere.'

It seems that the ban on multiple edges could be lifted in the enhanced UD without any ill consequences.[19]

4.3 Embedded coordination

A problem known to the UD community[20] is that there is no way to distinguish between embedded coordination, with the first conjunct itself being a coordinate structure, and flat coordination. There are about a dozen sentences in the Polish UD treebank described here where this is a potential problem, e.g.:

(43) Przewróciłem jakieś puszki, straciłem kamerę, ale
overturned.1SG.M some.ACC cans.ACC lost.1SG.M camera.ACC but
świeca płonie.
candle.NOM.SG.F burns.3SG
'I overturned some cans, lost my camera, but the candle still burns.'

In the LFG parsebank which is the input to the conversion procedure, this sentence is represented as a coordinate structure with the conjunction *ale* 'but'. The linearly first conjunct is also a coordinate structure (with comma acting as the conjunction) – see the f-structure in (44), where one of the two elements of the outer set representing the coordination is itself a two-element set representing the embedded coordination. This embedding of coordination cannot be directly represented in UD – see (45), which does not distinguish between flat ternary coordination and such binary coordination embedded within binary coordination.[21]

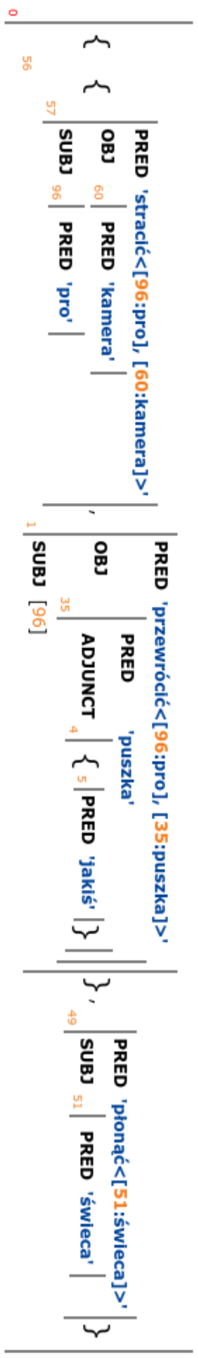
In practice, however, this is not a serious problem, as the right structure can usually – at least in the dozen or so cases in the current treebank – be inferred from the linear placement and kind of conjunctions. For example, a strictly binary contrastive conjunction *ale* is used in (43), so (45) cannot represent flat ternary coordination – it must represent embedded coordination. Moreover, if this turned out to be a serious practical problem, embedded coordination could be distinguished by subtyping the `conj` label, e.g. to `conj:coord`, whenever the head is a coordinate structure (as suggested to us by Nathan Schneider, p.c., August 2018); in the particular case of (45), the `conj` relation from *przewrócił* to *płonie* could be thus modified.

[19] As suggested by an anonymous reviewer, the possibilities created by lifting this ban may be constrained by well-formedness constraints on such co-occurrence of dependency edges.

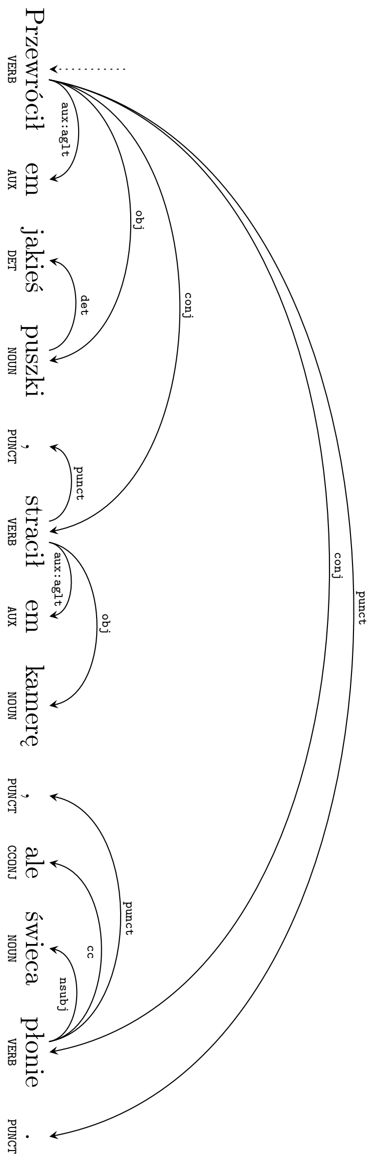
[20] `http://universaldependencies.org/u/dep/conj.html`

[21] Only the basic tree is shown here, as the enhanced representation is the same.

(44)


0
56
57
PRED 'stracić<[96:pro], [60:kamera]>'
OBJ 60 PRED 'kamera'
SUBJ 96 PRED 'pro'
1
PRED 'przewrócić<[96:pro], [35:puszka]>'
OBJ 35 PRED 'puszka'
ADJUNCT 4 5 PRED 'jakiś'
SUBJ [96]
49
PRED 'płonąć<[51:świeca]>'
SUBJ 51 PRED 'świeca'

(45)


Przewrócił VERB
em AUX
jakieś DET
puszki NOUN
, PUNCT
stracił VERB
em AUX
kamerę NOUN
, PUNCT
ale CCONJ
świeca NOUN
płonie VERB
. PUNCT
aux:aglt
obj
det
conj
punct
aux:aglt
obj
punct
cc
nsubj
conj
punct

4.4 Insufficient information in dependency labels

Much information is also lost because UD dependency labels are less informative than LFG attributes. For example, while LFG distinguishes between different kinds of oblique arguments (e.g., only in the f-structures given above: OBL, OBL-STR, OBL-INST, OBL-ADL, etc.), and distinguishes them from adjuncts, UD treats all such obliques and adjuncts alike, and marks them as obl. However, it is easy to extend UD in a way that makes representing such information possible. To this end, the mechanism of subtypes – already alluded to above (e.g., the relations expl:pv and expl:impers are subtypes of the general expl(etive) relation) – may be used. In fact, Zeman 2017 proposes to distinguish oblique arguments from adjuncts by subtyping the former to obl:arg, and similar subtypes may be used, e.g. to represent adlative oblique arguments as, say, obl:adl, etc.

The same mechanism may be used to re-introduce many other kinds of information currently lost in translation, including:

- the distinction between control and predicative complements, both marked in UD as xcomp (e.g. by subtyping the latter to xcomp:pred),
- the distinction between raising and control (e.g. by representing raising via xcomp:raising),
- the different grammatical functions of dependents of gerunds (now all broadly nominal dependents of gerunds are marked as nmod, but they could be subtyped to nmod:obj, nmod:obl, etc.),
- the distinction between semantic and non-semantic prepositions, e.g. by subtyping the case relation in the former to case:sem (and similarly for semantic and non-semantic complementisers),
- the distinction between eventuality and constituent negation (Przepiórkowski and Patejuk 2015), e.g. via the subtypes advmod:eneg and advmod:cneg; etc.[22]

Thus, the exercise described here shows that it is relatively easy to convert an LFG treebank into a full-blown enhanced UD representation. Surprisingly little information is lost in the conversion from LFG to enhanced UD and – as discussed in this section – many of the deficiencies of current UD are easy to rectify, and other surface rarely.

5 The LFG-based UD treebank of Polish

Texts in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ are drawn from two corpora: over 84% of the sentences come from the *National Corpus of Polish* (http://nkjp.pl/; Przepiórkowski et al.

[22] Since, in principle, any information present in f-structures may be preserved in such subtypes or as morphosyntactic features in the underlying CoNLL-U representation used in UD (see http://universaldependencies.org/format.html), and given the space constraints, we refrain from providing an exhaustive list of features preserved/lost in translation in the current conversion.

Table 1 Subcorpora of $\text{UD}^{\text{PL}}_{\text{LFG}}$

	trees	**tokens**
training	13,744	104,750
development	1745	13,105
test	1727	13,112

2011, 2012) and almost 16% – from the *Corpus of 1960s Polish* (`http://clip.ipipan.waw.pl/PL196x`; Kurcz et al. 1990, Bień and Woliński 2003, Ogrodniczuk 2003). Both corpora were lemmatised and morphosyntactically tagged, and these lemmata and tags are almost always preserved in $\text{UD}^{\text{PL}}_{\text{LFG}}$.

More directly, the sentences in $\text{UD}^{\text{PL}}_{\text{LFG}}$ come from the LFG treebank of Polish (Patejuk and Przepiórkowski 2014b). 19,597 sentences with their LFG syntactic structures form an input to the conversion described above. Many of these are sentences with multiple possible LFG analyses, as well as accidental duplicates. After conversion, only unique UD structures are retained, i.e., a sentence may appear in the corpus a couple of times only with different dependency annotations (ideally reflecting genuine ambiguities, difficult to disambiguate by human annotators). As a result, the $\text{UD}^{\text{PL}}_{\text{LFG}}$ treebank contains 17,246 dependency representations (with 130,967 segments) for 17,190 different sentences.

These 17,246 representations were split into training, development and test subcorpora in two stages, in compliance with UD guidelines.[23] First, for each sentence, it was checked whether this sentence occurs in the previous UD treebank of Polish, $\text{UD}^{\text{PL}}_{\text{SZ}}$. If it occurred in the training corpus there, it was also assigned – with all its dependency structures, if there were more than one – to the training subcorpus of the $\text{UD}^{\text{PL}}_{\text{LFG}}$ treebank. Otherwise, if it was found in the $\text{UD}^{\text{PL}}_{\text{SZ}}$ development corpus, it was assigned to the $\text{UD}^{\text{PL}}_{\text{LFG}}$ development corpus. Otherwise, if it occurred in the $\text{UD}^{\text{PL}}_{\text{SZ}}$ test corpus, it was assigned to the $\text{UD}^{\text{PL}}_{\text{LFG}}$ test corpus. Altogether, 3502 (2594 + 439 + 469, respectively) dependency representations were pre-classified to the three subcorpora this way.

Second, the remaining sentences were randomly added to the development and test subcorpora until each of these subcorpora contained more than 20% of the whole corpus, in terms of both the number of dependency representations and the number of tokens. The rest of the sentences were added to the training corpus. This procedure resulted in the split summarised in Table 1.

About 42.1% of the sentences represent the fiction genre, 39.1% – news, 7.4% – nonfiction, 7.3% – spoken, 3% – interactive Internet texts (forums, chatrooms, etc.), and there are also traces of static Internet pages (0.8%), academic style (0.3%) and legal texts (0.1%). For each sentence, genre is explicitly given in a comment to the sentence. In the case of sentences derived from the National Corpus of Polish, this genre information is taken directly from the headers of appropriate texts; in the case of sentences from the Corpus of 1960s

[23] `http://universaldependencies.org/release_checklist.html#data-split`

Polish, they were derived from two (of five) parts of the corpus, News and Fiction, and were classified accordingly.

6 Comparison with the previous UD treebank of Polish

$\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ is the first Polish UD treebank making use of enhanced dependencies. It has been available since February 2018 and it was officially released as part of UD version 2.2 in July 2018. However, there is also another UD treebank of Polish, available since UD release 1.2 in November 2015, namely, $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$. That treebank is based on the *Składnica zależnościowa* treebank (Wróblewska 2014; `http://zil.ipipan.waw.pl/Składnica`) version 0.5, which is the result of automatic conversion from a constituency parsebank *Składnica* (Świdziński and Woliński 2010). *Składnica zależnościowa* was first converted – by Dan Zeman and colleagues – to the Prague dependency style and then, from this format, to Universal Dependencies (HamleDT 3.0, 2015; Zeman et al. 2014).[24] The rest of this section briefly compares mid-2018 versions of these two Polish UD treebanks.

6.1 Tokenisation

There are at least two tokenisation differences between the two treebanks. First, $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$, but not $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$, takes advantage of the possibility to represent sequences of tokens written without intervening spaces also as single tokens, as in *Straciłem równowagę.* 'I lost my balance', lit. 'lost.1sg.m balance.acc.sg.f', where *Straciłem* 'lost.1sg.m' is a sequence of two tokens: *Stracił* 'lost.sg.m' and *em* 'aux.1sg'. In the CoNLL-U representation of this sentence, there are five lines (apart from the comment lines) in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$; (46) shows the first four columns and the final column (with omitted material between them indicated by '. . . '):

```
(46) 1-2 Straciłem _          _      ...  _
     1   Stracił   stracić    VERB   ...  _
     2   em        być        AUX    ...  _
     3   równowagę równowaga  NOUN   ...  SpaceAfter=No
     4   .         .          PUNCT  ...  _
```

On the other hand, the partial representation of the same sentence in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ is as in (47) – it differs not only in the lack of one line, but also in the more consistent – in our opinion – use of the `SpaceAfter=No` feature.

```
(47) 1   Stracił   stracić    VERB   ...  SpaceAfter=No
     2   em        być        AUX    ...  _
     3   równowagę równowaga  NOUN   ...  SpaceAfter=No
     4   .         .          PUNCT  ...  _
```

[24] See the description of $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ at `https://github.com/UniversalDependencies/UD_Polish-SZ/blob/dev/README.md`.

In the case of Polish, both representations give exactly the same information and may be easily converted one to another.

The second – minor – difference is that $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ does not indicate the lack of space between a preposition and the following short pronominal form, as in *doń* 'to him(/it/her)' – neither via the `SpaceAfter=No` feature, nor via an additional line for such a multi-token unit. This error should be easy to correct in future releases of $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$.

6.2 Morphosyntax

There are various morphosyntactic differences between the two treebanks; some – discussed immediately below – stem from some controversial decisions taken by the developers of $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$, others are probably the result of lack of certain kinds of information in the input data converted to $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$, and still others are minor errors, which should be easy to correct in future editions of $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$.

Polish has five genders (Mańczak 1956), including 3 masculine genders sometimes – misleadingly – called 'human masculine', 'animate masculine' and 'inanimate masculine'. There are good morphosyntactic tests making it possible to distinguish the three (sub)genders, without any recourse to semantic intuition. Since the correlation between the three masculine genders and semantic animacy is far from perfect, these masculine genders are distinguished in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ via the values of the new `SubGender` feature (a solution suggested to us by Dan Zeman, p.c.). In $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$, however, the `Animacy` feature is employed to this end, with three possible values: `Hum` for 'human masculine', `Nhum` for 'animate masculine' and `Inan` for 'inanimate masculine'. This is highly misleading – the cursory inspection of the 150 lemmata whose forms are marked as 'animate masculine' `NOUN`s in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ suggests that perhaps only about half of them refer to animals. For example, considering such lemmata starting in T, only two out of seven – TRZMIEL 'bumblebee' and TYGRYS 'tiger' – are semantically animate:

– TAROT – 'tarot',
– TENIS – 'tennis',
– TIR – a heavy vehicle,
– TRUP – 'corpse',
– TRZECI – 'third' (possibly an error in input data),
– TRZMIEL – 'bumblebee',
– TYGRYS – 'tiger'.

A closely related problem stems from the lack of proper handling of 'derogatory' forms of 'human masculine' nouns in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$, e.g. *profesory* 'professors (derogatory)' vs. the neutral *profesorowie*. Such forms behave morphosyntactically as if they were 'animate masculine', so the value of their `Animacy` feature is `Nhum` in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$, even though they are without exception semantically human masculine. (This problem is statistically insignificant, though, as it only concerns four tokens.) In $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ such derogatory forms are marked as `Polite=Depr`.

Another controversial decision – or perhaps simply a conversion error – is the annotation of morphologically impersonal *-no/-to* forms as adjectival passive participles in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$, i.e., as tokens with the `ADJ` coarse part of speech and with `VerbForm=Part` and `Voice=Pass`, as well as, somewhat curiously, `Case=Nom`, `Gender=Neut` and `Number=Sing` among their features. Tokens such as *wyrzucano* 'one used to throw away' or *zdobyto* 'one conquered', are – uncontroversially – purely verbal,[25] with no grammatical case, no clear values of number and gender, and they may be formed from verbs which do not passivise at all. In $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ they are treated as finite verbs with the distinguishing feature `Person=0` marking their morphologically impersonal status.

Three other differences probably stem from the lack of appropriate information in the data that was used to develop $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$. First, $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ does not distinguish between relative and interrogative uses of such (broadly understood) pronouns as KTO 'who', CO 'what' and KTÓRY 'which', marking them all as `PronType=Int,Rel`, i.e., as 'interrogative or relative'. In contrast, such pronouns are appropriately marked as interrogative or as relative, i.e., they are disambiguated in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$.

Second, the UD coarse part of speech tag `X`, "used for words that for some reason cannot be assigned a real part-of-speech category",[26] is used in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ in two situations. One is easy to correct (as well as rare) and concerns predicative-only (short) adjectives – such forms are in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ tagged as `ADJ` and assigned the `Variant=Short` feature. The other concerns 273 tokens (with 46 different lemmata) of abbreviations. Such abbreviations are tagged with specific parts of speech (in morphosyntactic features) in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$, but only as `X` in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$.

Third, last and certainly least, $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ does not distinguish between prepositions and postpositions, marking them all as `AdpType=Prep`. But as there is only one clear exception to the generalisation that Polish adpositions are always prepositions, namely, the postposition TEMU 'ago', this only affects 28 tokens representing this lemma.

6.3 Syntax

The fundamental difference between $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ and $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ is the presence of enhanced dependencies in the latter. The intensive use of secondary edges in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ makes it possible to express many syntactic relations absent in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$, including grammatical control and sharing of dependents between conjuncts in coordinate structures.

Apart from this, probably the biggest conceptual difference between the two UD treebanks of Polish concerns the *argument–adjunct distinction*, as well

[25] See e.g. Blevins 2003, Lavine 2004 and references therein. Unlike in the case of Polish, Ukrainian *-no/-to* forms do exhibit certain properties of passive participles and should perhaps be analysed as such.

[26] `http://universaldependencies.org/u/pos/X.html`

as the definition of direct and indirect objects. $\text{UD}^{\text{PL}}_{\text{LFG}}$ attempts to follow the general UD philosophy of not trying to distinguish arguments from adjuncts:[27]

> The UD taxonomy is centered around the fairly clear distinction between core arguments (subjects, objects, clausal complements) versus other dependents. It does not make a distinction between adjuncts (general modifiers) versus oblique arguments (arguments said to be selected by a head but not expressed as a core argument).[28]

Following this philosophy, nominal core arguments (subjects, direct and indirect objects) are defined in $\text{UD}^{\text{PL}}_{\text{LFG}}$ in a narrow and linguistically well-founded way, with the effect that many broadly nominal dependents – both bare and prepositional – which would traditionally be classified as complements (i.e., arguments) are not distinguished from traditional nominal adjuncts.

On the other hand, $\text{UD}^{\text{PL}}_{\text{SZ}}$ reintroduces the argument–adjunct distinction: apart from defining objects in a very broad and somewhat inconsistent way (see below), it also splits the oblique dependents into arguments, marked as `obl:arg`, and adjuncts, marked as `obl` (without any explicit subtype). The proposal to re-introduce argument–adjunct distinction into UD is explicitly presented in Zeman 2017.

A related important difference is the definition of *direct objects*, marked as `obj`. In $\text{UD}^{\text{PL}}_{\text{LFG}}$, direct object is defined in a precise and at the same time traditional (e.g. Gołąb et al. 1968: 132, Urbańczyk 1992: 62) way as that dependent of a verb which is realised as the subject in passive occurrences of this verb. On the other hand, in $\text{UD}^{\text{PL}}_{\text{SZ}}$ the label `obj` is used for all (non-subject) bare nominal arguments, whether they passivise or not. Given that there are also bare nominal adjuncts in Polish, this definition of direct objects again presupposes the argument–adjunct distinction. Also, $\text{UD}^{\text{PL}}_{\text{SZ}}$ treats subcategorised clauses, marked as `ccomp`, as direct objects. Since there is a ban on two direct object dependents of a single verb, the situation where one verb has a `ccomp` dependent and an `obj` dependent is not allowed – as discussed immediately below, the direct object is then re-analysed as an indirect object.

Also the definitions of *indirect objects*, `iobj`, differ in the two treebanks, although neither is optimal. In $\text{UD}^{\text{PL}}_{\text{LFG}}$, indirect objects are defined as subcategorised bare dative (non-passivisable) dependents; the subcategorisation requirement re-introduces – albeit in a very limited way – the argument–adjunct dichotomy. While such limited references to this dichotomy are present also elsewhere in the UD standard, this goes against the spirit of UD and should be changed in future editions of $\text{UD}^{\text{PL}}_{\text{LFG}}$; since traditional Polish grammars do not recognise the class of indirect objects, perhaps all `iobj` labels should simply be replaced by `obl` labels. The definition of indirect objects in $\text{UD}^{\text{PL}}_{\text{SZ}}$ is even more questionable: if there are two candidates for the direct object dependency, only one is assigned the `obj` label. In particular, if a subcategorised clause is one of the two candidates, it is assigned the status of direct object and the bare

[27] See Przepiórkowski and Patejuk 2018 on inconsistencies in the current version of UD regarding the argument–adjunct distinction and the core–obliqueness distinction.

[28] `http://universaldependencies.org/u/overview/syntax.html`

accusative dependent receives the `iobj` label. This leads to some annotations which are in direct conflict with linguistically motivated definitions of direct objects. For example, in (48) (sentence `train-s2613` in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$), the verb *spytało* 'asked' combines with the numeral subject *kilka osób* 'several people', the accusative nominal *mnie* 'me' and the subordinate clause *czy jestem...* 'whether I am...'; since the subordinate clause is subcategorised, it is marked as `ccomp`, but in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ that means that *mnie* 'me' must be marked as indirect object, `iobj`, even though it becomes the subject under passivisation and it occurs in the accusative case, so it is a prototypical direct object.

(48) Kilka osób spytało mnie, czy jestem dzięki feminizmowi
several people asked me.ACC.SG whether am.1SG thanks feminism.DAT
szczęśliwsza.
happier.NOM.SG.F
'Some people have asked me whether feminism made me happier.'

This leads to inconsistencies in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$, as in other sentences, lacking such subordinate clause dependents, analogous accusative dependents are correctly marked as direct objects (`obj`), as is the case with *ją* 'her' in (49) (sentence `train-s2739` in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$):

(49) Chciał ją spytać o wiele rzeczy.
wanted her.ACC ask.INF about many things
'He wanted to ask her about many things.'

Since *mnie* 'me' in (48) and *ją* 'her' in (49) bear exactly the same semantic role with respect to the two forms of the verb SPYTAĆ 'ask' and have the same grammatical properties (passivisability, grammatical case, etc.), this is a clear case of intra-linguistic annotation inconsistency.

In summary, we believe that the approach to grammatical functions in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ is both more consistent intra-linguistically and more justified linguistically than in the case of $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$.

It is perhaps worth noting that, because of the fundamental differences discussed in this section, especially the lack of enhanced dependencies modelling dependent-sharing in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ and very different approaches to direct and indirect objects in the two treebanks, as well as due to the lack of certain kinds of morphosyntactic information in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ pointed out in the previous section, $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ cannot easily be converted to the $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ schema.

6.4 Underlying data

The ultimate source of texts and original morphosyntactic information in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ is the 1-million-word manually annotated subcorpus of the *National Corpus of Polish*, which is also the source of almost 85% of texts in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$. This means that the values of the XPOS field of the CoNLL-U representation used in UD are taken from the same tagset, but – given that some morphosyntactic analyses were modified in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ – not that they would necessarily be

Table 2 Quantitative comparison of $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ and $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$

	$\mathbf{UD}^{\mathrm{PL}}_{\mathbf{SZ}}$	$\mathbf{UD}^{\mathrm{PL}}_{\mathbf{LFG}}$
sentences (running)	8227	17,246
sentences (different)	8139	17,190
tokens (running)	84,316	130,967
lemmata (different)	13,688	15,797

identical for the same sentence in the two treebanks. For example, typical (non-agreeing) numerals in the subject position are marked as nominative in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ but as accusative in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$, in accordance with the analysis of such subjects in Przepiórkowski 1999, 2004 (and following some earlier observations, including Małecki 1863: 297 and Franks 1995: 139).

The sizes of the two treebanks are compared in Table 2. $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ is much larger: it contains 17,246 running sentences (17,190 types; duplicate sentences have different analyses), compared to 8227 running sentences in $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ (8139 types; duplicate sentences may have the same analyses). In terms of running tokens, the respective numbers are 130,967 ($\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$) vs. 84,316 ($\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$), which implies that $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$ sentences are longer on the average. $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ is also a little richer lexically (which is to be expected, given the bigger size).

7 Conclusion

It is well known that simple dependency trees are not sufficiently expressive to adequately represent such important linguistic information as grammatical control or dependent-sharing in coordinate structures. Theoretical dependency approaches attempt to compensate for such expressive deficiencies either by assuming representations going beyond dependency trees (e.g. Tesnière's 1959, 2015 Dependency Syntax or Hudson's 1984, 2010 Word Grammar) or by postulating multiple levels of representation, each of which might be a simple tree (e.g. Mel'čuk's 1988, 2009 Meaning–Text Theory or the Praguian Functional Generative Description of Sgall et al. 1986). The current version 2.2 of the Universal Dependencies standard for treebank annotation adopts both approaches: it assumes two representations, one of which is a dependency graph, rather than a simple dependency tree.

Given this richer expressive power of enhanced UD, the natural question is to what extent information may be preserved in the conversion from full-fledged linguistic annotations offered by theories such as Lexical Functional Grammar. While we have not performed a formal and exhaustive comparison of the two annotation schemata, we approached this question from a practical perspective, by performing a conversion of an existing LFG treebank to UD. The conclusion is that relatively little information is lost in translation. Moreover, the reasons for this loss are not fundamental to the UD approach, and may easily be rectified in future versions of this standard.

The resulting treebank, $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$, is one of the first treebanks to make extensive use of such enhanced representations. Also, it is considerably larger than the previous UD treebank of Polish, $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{SZ}}$, and it is free from a number of deficiencies present in that previous treebank. An accompanying monograph – Patejuk and Przepiórkowski 2018 – documents the input LFG treebank of Polish, describes the conversion procedure in minute detail, and presents the resulting UD treebank in a more meticulous way. We hope that $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ – available directly from `http://universaldependencies.org/` and searchable via the INESS infrastructure at `http://clarino.uib.no/iness/` (where it is called `pol-ud-lfg-2.2-dep`) – will turn out to be useful both for natural language processing applications, and in linguistic research.

Acknowledgements We would like to cordially thank Joakim Nivre and Dan Zeman for their infinite patience in answering a myriad of diverse UD-related questions during the development of the $\mathrm{UD}^{\mathrm{PL}}_{\mathrm{LFG}}$ treebank, and the two anonymous LREV reviewers for their comments, which led to some important improvements. The paper also benefitted from the comments of audiences at CAS in Oslo (see below), at the 23rd International Lexical Functional Grammar Conference 2018 in Vienna, and at the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018) at COLING 2018 in Santa Fe, NM, where it was presented as a keynote talk. The conversion of the LFG treebank into UD was partially supported by the Polish Ministry of Science and Higher Education within the *CLARIN ERIC* programme 2016–2018 (`http://clarin.eu/`), as well as within the *Mobilność Plus* mobility grant awarded to Agnieszka Patejuk. This article was written during the authors' fellowship at the Oslo Center for Advanced Study (CAS) at the Norwegian Academy of Science and Letters (`https://cas.oslo.no/`), within the group "SynSem: From Form to Meaning – Integrating Linguistics and Computing" led by Dag Haug and Stephan Oepen. It is very possible that, if not for our involvement in CAS, neither this article, nor the UD treebank of Polish presented here, would ever see the light of day.

References

Bień JS, Woliński M (2003) Wzbogacony korpus *Słownika frekwencyjnego polszczyzny współczesnej*. In: Linde-Usiekniewicz J, Huszcza R (eds) Prace językoznawcze dedykowane Profesor Jadwidze Sambor, Uniwersytet Warszawski, Wydział Polonistyki, Warsaw, pp 6–10

Blevins JP (2003) Passives and impersonals. Journal of Linguistics 39(3):473–520

Bresnan J (ed) (1982) The Mental Representation of Grammatical Relations. The MIT Press, Cambridge, MA

Butt M, King TH (eds) (2015) The Proceedings of the LFG'15 Conference, CSLI Publications, Stanford, CA, URL `http://cslipublications.stanford.edu/LFG/20/lfg15.html`

Çetinoğlu Ö, Foster J, Nivre J, Hogan D, Cahill A, van Genabith J (2010) LFG without c-structures. In: Dickinson M, Müürisep K, Passarotti M (eds) Proceedings of the Ninth International Workshop on Treebanks and Linguistic Theories (TLT 9), Tartu, Estonia, pp 43–54

Dalrymple M (2001) Lexical Functional Grammar. Academic Press, San Diego, CA

Forst M (2003) Treebank conversion: Creating a German f-structure bank from the TIGER corpus. In: Butt M, King TH (eds) The Proceedings of the LFG'03 Conference, CSLI Publications, Stanford, CA, pp 205–216

Franks S (1995) Parameters of Slavic Morphosyntax. Oxford University Press, New York

van Genabith J, Way A, Sadler L (1999) Semi-automatic generation of f-structures from treebanks. In: Butt M, King TH (eds) The Proceedings of the LFG'99 Conference, CSLI Publications, University of Queensland, Brisbane, URL `https://web.stanford.edu/group/cslipublications/cslipublications/LFG/LFG4-1999/`

Gołąb Z, Heinz A, Polański K (1968) Słownik terminologii językoznawczej. Wydawnictwo Naukowe PWN

Haug D (2012) From dependency structures to LFG representations. In: Butt M, King TH (eds) The Proceedings of the LFG'12 Conference, CSLI Publications, Stanford, CA, pp 271–291, URL `http://cslipublications.stanford.edu/LFG/17/lfg12.html`

Hudson R (1984) Word Grammar. Blackwell, Oxford

Hudson R (2010) An Introduction to Word Grammar. Cambridge University Press

Johannessen JB (1996) Partial agreement and coordination. Linguistic Inquiry 27(4):661–676

Kurcz I, Lewicki A, Sambor J, Szafran K, Woronczak J (1990) Słownik frekwencyjny polszczyzny współczesnej. Wydawnictwo Instytutu Języka Polskiego PAN, Cracow

Lavine JE (2004) The morphosyntax of Polish and Ukrainian *-no/-to*. Journal of Slavic Linguistics 13(1):75–117

Małecki A (1863) Gramatyka języka polskiego większa. Lwów

Mańczak W (1956) Ile jest rodzajów w polskim? Język Polski XXXVI(2):116–121

Mel'čuk I (1988) Dependency Syntax: Theory and Practice. The SUNY Press, Albany, NY

Mel'čuk I (2009) Dependency in natural language. In: Polguère A, Mel'čuk I (eds) Dependency in Linguistic Description, John Benjamins, pp 1–110

Meurer P (2017) From LFG structures to dependency relations. In: Rosén V, Smedt KD (eds) The Very Model of a Modern Linguist, Bergen Language and Linguistics Studies, vol 8, University of Bergen Library, Bergen, pp 183–201, URL `https://bells.uib.no/index.php/bells/article/view/1341`

Munn AB (1993) Topics in the syntax and semantics of coordinate structures. Ph.D. Thesis, University of Maryland

Nivre J, de Marneffe MC, Ginter F, Goldberg Y, Hajič J, Manning CD, McDonald R, Petrov S, Pyysalo S, Silveira N, Tsarfaty R, Zeman D (2016) Universal Dependencies v1: A multilingual treebank collection. In: Calzolari N, Choukri K, Declerck T, Grobelnik M, Maegaard B, Mariani J, Moreno A, Odijk J, Piperidis S (eds) Proceedings of the Tenth International Conference on Language Resources and Evaluation, LREC 2016, ELRA, European Language Resources Association (ELRA), Portorož, Slovenia, pp 1659–1666, URL http://www.lrec-conf.org/proceedings/lrec2016/pdf/348_Paper.pdf

Ogrodniczuk M (2003) Nowa edycja wzbogaconego korpusu słownika frekwencyjnego. In: Gajda S (ed) Językoznawstwo w Polsce. Stan i perspektywy, Komitet Językoznawstwa, Polska Akademia Nauk oraz Instytut Filologii Polskiej, Uniwersytet Opolski, Opole, pp 181–190, URL http://bc.klf.uw.edu.pl/104/

Øvrelid L, Kuhn J, Spreyer K (2009) Cross-framework parser stacking for data-driven dependency parsing. TAL 50(3):109–138

Patejuk A, Przepiórkowski A (2012) Towards an LFG parser for Polish: An exercise in parasitic grammar development. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, ELRA, Istanbul, Turkey, pp 3849–3852, URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/150_Paper.pdf

Patejuk A, Przepiórkowski A (2014a) Structural case assignment to objects in Polish. In: Butt M, King TH (eds) The Proceedings of the LFG'14 Conference, CSLI Publications, Stanford, CA, pp 429–447, URL http://web.stanford.edu/group/cslipublications/cslipublications/LFG/19/papers/lfg14patprz2.pdf

Patejuk A, Przepiórkowski A (2014b) Synergistic development of grammatical resources: A valence dictionary, an LFG grammar, and an LFG structure bank for Polish. In: Henrich V, Hinrichs E, de Kok D, Osenova P, Przepiórkowski A (eds) Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT 13), Department of Linguistics (SfS), University of Tübingen, Tübingen, pp 113–126, URL http://nlp.ipipan.waw.pl/Bib/pat:prz:14:synergy.pdf

Patejuk A, Przepiórkowski A (2015a) An LFG analysis of the so-called reflexive marker in Polish. In: Butt and King (2015), pp 270–288, URL http://web.stanford.edu/group/cslipublications/cslipublications/LFG/20/papers/lfg15patprz.pdf

Patejuk A, Przepiórkowski A (2015b) Parallel development of linguistic resources: Towards a structure bank of Polish. Prace Filologiczne LXV:255–270

Patejuk A, Przepiórkowski A (2018) From Lexical Functional Grammar to Enhanced Universal Dependencies: Linguistically informed treebanks of Polish. Institute of Computer Science, Polish Academy of Sciences, Warsaw, URL http://nlp.ipipan.waw.pl/Bib/pat:prz:18:book.pdf

Przepiórkowski A (1999) Case assignment and the complement-adjunct dichotomy: A non-configurational constraint-based approach. Ph.D. Thesis, Universität Tübingen, Tübingen, URL http://nlp.ipipan.waw.pl/~adamp/Dissertation/

Przepiórkowski A (2004) O wartości przypadka podmiotów liczebnikowych. Biuletyn Polskiego Towarzystwa Językoznawczego LX:133–143, URL http://nlp.ipipan.waw.pl/~adamp/Papers/2003-bptj-case/

Przepiórkowski A, Patejuk A (2015) Two representations of negation in LFG: Evidence from Polish. In: Butt and King (2015), pp 322–336, URL http://web.stanford.edu/group/cslipublications/cslipublications/LFG/20/papers/lfg15przpat.pdf

Przepiórkowski A, Patejuk A (2018) Arguments and adjuncts in Universal Dependencies. In: Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018), Santa Fe, NM, pp 3837–3852, URL http://aclweb.org/anthology/C18-1324

Przepiórkowski A, Bańko M, Górski RL, Lewandowska-Tomaszczyk B, Łaziński M, Pęzik P (2011) National Corpus of Polish. In: Vetulani Z (ed) Proceedings of the 5th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, Poznań, Poland, pp 259–263

Przepiórkowski A, Bańko M, Górski RL, Lewandowska-Tomaszczyk B (eds) (2012) Narodowy Korpus Języka Polskiego. Wydawnictwo Naukowe PWN, Warsaw, URL http://nkjp.pl/settings/papers/NKJP_ksiazka.pdf

Rosén V, De Smedt K, Meurer P, Dyvik H (2012) An open infrastructure for advanced treebanking. In: LREC 2012 META-RESEARCH Workshop on Advanced Treebanking, ELRA, Istanbul, Turkey, pp 22–29

Saloni Z, Świdziński M (1985) Składnia współczesnego języka polskiego, 2nd edn. Wydawnictwo Naukowe PWN, Warsaw

Sgall P, Hajičová E, Panevová J (1986) The Meaning of the Sentence in Its Semantic and Pragmatic Aspects. Reidel, Dordrecht

Sulger S, Butt M, King TH, Meurer P, Laczkó T, Rákosi G, Dione CB, Dyvik H, Rosén V, De Smedt K, Patejuk A, Çetinoğlu Ö, Arka IW, Mistica M (2013) ParGramBank: The ParGram parallel treebank. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Sofia, Bulgaria, pp 550–560

Świdziński M, Woliński M (2010) Towards a bank of constituent parse trees for Polish. In: Sojka P, Horák A, Kopeček I, Pala K (eds) Text, Speech and Dialogue: 13th International Conference, TSD 2010, Brno, Czech Republic, Springer-Verlag, Heidelberg, Lecture Notes in Artificial Intelligence, vol 6231, pp 197–204

Tesnière L (1959) Éléments de Syntaxe Structurale. Klincksieck, Paris

Tesnière L (2015) Elements of Structural Syntax. John Benjamins, Amsterdam

Urbańczyk S (ed) (1992) Encyklopedia języka polskiego. Ossolineum, Wrocław

Wróblewska A (2014) Polish dependency parser trained on an automatically induced dependency bank. Ph.D. Thesis, Institute of Computer Science, Polish Academy of Sciences, Warsaw

Zeman D (2017) Core arguments in Universal Dependencies. In: Montemagni S, Nivre J (eds) Proceedings of the Fourth International Conference on Dependency Linguistics (DepLing 2017), Pisa, Italy, pp 287–296

Zeman D, Dušek O, Mareček D, Popel M, Ramasamy L, Štěpánek J, Žabokrtský Z, Hajič J (2014) HamleDT: Harmonized multi-language dependency treebank. Language Resources and Evaluation 48(4):601–637

Zhang NN (2009) Coordination in Syntax. Cambridge University Press, Cambridge