

Experiments in PCFG-like Disambiguation of Constituency Parse Forests for Polish

Marcin Woliński^(✉) and Dominika Rogozińska

Institute of Computer Science, Polish Academy of Sciences, Warszawa, Poland
wolinski@ipipan.waw.pl

Abstract. The work presented here is the first attempt at creating a probabilistic constituency parser for Polish. The described algorithm disambiguates parse forests obtained from the Świgrą parser in a manner close to Probabilistic Context Free Grammars. The experiment was carried out and evaluated on the Składnica treebank. The idea behind the experiment was to check what can be achieved with this well known method. Results are promising, the approach presented achieves up to 94.1 % PARSEVAL F-measure and 92.1 % ULAS. The PCFG-like algorithm can be evaluated against existing Polish dependency parser which achieves 92.2 % ULAS.

1 Motivation and Context

The main incentive for the present work is the availability of the Składnica treebank of Polish (Woliński et al. 2011; Świdziński and Woliński 2010)¹, which for the first time provides the means to attempt probabilistic parsing of Polish. Składnica is a constituency treebank based on parse forests generated by the Świgrą parser and subsequently disambiguated by annotators.

The parser generates parse forests representing all possible parse trees for a given sentence. Then the correct tree is marked in the forest by annotators.

Including a probabilistic module in the parsing process of Świgrą would require tight integration and deep insight into its workings. Therefore, for the present experiments we have taken an approach that is technically simpler. We generate complete forests with unchanged Świgrą and then the probabilistic algorithm has to select one of the generated trees. This way the algorithm solves exactly the same problem as annotators of the training corpus.

In this paper we present a series of experiments based on Probabilistic Context Free Grammars as a method for assigning probabilities to parse trees.

2 Scoring the Results

For evaluating disambiguated parses we use the PARSEVAL precision and recall measures (Abney et al. 1991), which count correctly recognised phrases in the

¹ <http://zil.ipipan.waw.pl/Skladnica>.

algorithm output. A phrase, represented in the constituency tree by an internal node, is correct iff it has the right non-terminal and spans the correct fragment of the input text (it has the correct yield).

Precision and recall is computed across the whole set of sentences being processed:

$$\text{Precision} = \frac{\text{number of correct nodes}}{\text{number of nodes selected by the algorithm}}$$

$$\text{Recall} = \frac{\text{number of correct nodes}}{\text{number of nodes in training trees}}$$

In all experiments described below the values of precision and recall are close to each other (within 1 % point). This is not very surprising: the trees selected by the algorithms are close in the number of nodes to the training trees. So usually when a node is selected that should not be (spoiling precision), some of the nodes that should be selected is not (spoiling recall). For that reason we present the results in the aggregated form of F-measure (harmonic mean of precision and recall).

Non-terminals in Składnica are complex terms. The label of a nonterminal unit (e.g., *nominal phrase fno*) is accompanied by several attributes (10 in the case of *fno*: morphological features such as case, gender, number, and person, as well as a few attributes specific to the grammar in use). We provide two variants of F-measures: taking into account only whether the labels of non-terminal units match – reported as F_L or requiring a match on all attributes – F_A .

We count the measures against internal nodes of the trees only, that is non-terminals. The terminals, carrying morphological interpretations of words, are unambiguous in the manually annotated corpus.

Składnica contains information about heads of phrases, which makes it easy to convert constituency trees to (unlabelled) dependency trees. We perform such a conversion to count *unlabelled attachment score* (ULAS, the ratio of correctly assigned dependency edges) for resulting trees. This allows us to compare our results with those of Wróblewska and Woliński (2012). We do not use Wróblewska’s procedure for converting the trees to labelled dependency trees since it contains some heuristic elements that could influence the results.

In all the reported experiments ten-fold cross validation was used. Składnica contains trees for about 8000 sentences. This set was randomly divided into ten parts. In each of ten iterations nine parts were used for building the model and the remaining one to evaluate it.

3 Monkey Dendrologist – The Baseline

For the baseline of our experiments we have selected the following model. The task at hand mimics the work of annotators (called dendrologists by the authors of Składnica), so for the baseline we want to mimic a dendrologist who performs disambiguation by taking random decisions at each step.

In a shared parse forest typically only some nodes are ambiguous. These nodes have more than one decomposition into smaller phrases in the tree. This situation corresponds to the possibility of using more than one grammar rule to obtain the given node. Disambiguation can be seen as deciding for each ambiguous node which rule to take.

In the tree in Fig. 1 ambiguous nodes are marked with rows of tiny rectangles with arrows (which allow to select various realisations in the search tool of Składnica). Each rectangle represents one realisation of the given node. In this tree 5 of 35 internal nodes are ambiguous.

A “monkey dendrologist” considers the ambiguous nodes starting from the root of the tree and for each of them selects with equal probabilities one of possible realisations. Note that these decisions are not independent: selecting a realisation for a node determines the set of ambiguous nodes that have to be considered in its descendant nodes. Ambiguous nodes that lay outside of these selected subtrees will not even be considered.

A variant of monkey dendrologist is a “mean monkey dendrologist”. This one when considering a node first checks in the reference treebank which variant is correct and then selects randomly from the other variants.

The following table presents disambiguation quality of monkey dendrologists:

	F_L	F_A	ULAS
Mean monkey	0.859	0.696	0.808
Monkey	0.877	0.759	0.832

For some sentences Świgr generates very many parses, giving the impression that every structure is possible. Nonetheless, the above numbers show that the rules of the grammar limit possible trees quite strongly. The F_A score for the dendrologist that deliberately chooses wrong shows that about 70% of the nodes are unambiguous.

4 PCFG-like Disambiguation

The idea of Probabilistic Context Free Grammars is to associate probabilities with rules of a context free grammar. Applications of rules are considered independent, and so the probability of a given parse tree is computed as a product of probabilities of all rules used.

Probabilities of rules in PCFG are estimated probabilities of a given non-terminal being rewritten into a given sequence of non-terminals (that is probability of a given sequence of non-terminals to become the children of a given non-terminal). This is counted on a treebank by dividing the number of times a given rule was applied by the number of times all rules with the same left hand side were applied.

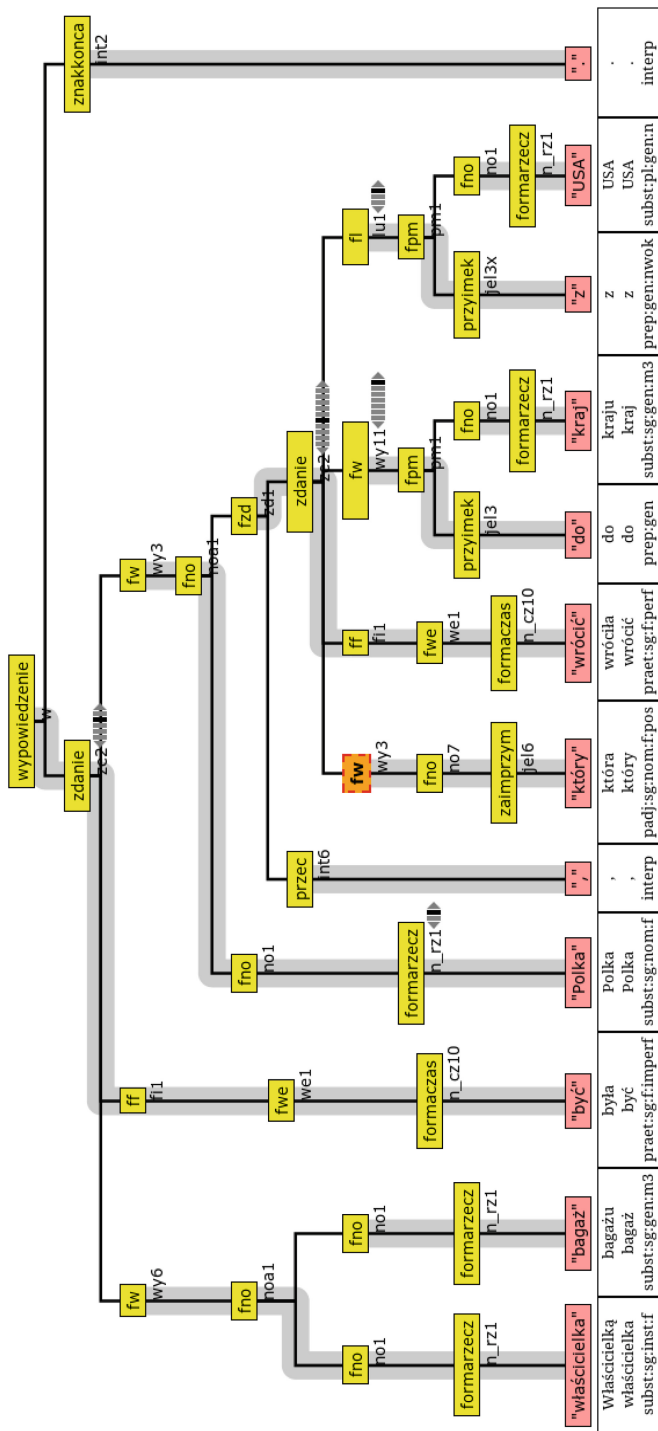


Fig. 1. A Składnica tree for the sentence
 Właścicielką bagażu była Polka, która wróciła do kraju z USA.
 owner luggage was Pole which returned to country from U.S.
 ‘The owner of the luggage was a Pole who returned to the country from the U.S.’

The grammar of Świgr is a Definite Clause Grammar (Pereira and Warren 1980) with an extension allowing its CFG-like rules to include optional and repeatable elements in their right hand sides. This means a single rule can generate nodes of various arities in the trees, which makes assigning probabilities to rules doubtful. Nonetheless this idea can be applied to Składnica trees by assigning probabilities to couples ⟨parent, list of children⟩. In other words, we try to estimate the probability of a given node having a given sequence of nodes as its children.

The algorithm operates on packed (shared) parse forests (Billot and Lang 1989), whose nodes are polynomial in number, even if they represent an exponential number of trees. The key point in effective processing is to construct scores over the trees without constructing all separate trees.

The disambiguation algorithm computes probabilities using a dynamic procedure. The goal is to find the most probable parse tree. As we are maximizing a product, in each ambiguous node (constituent) we can choose the realization with the highest PCFG probability. We perform the computation in a bottom-up manner, which allows us to avoid producing and processing all possible parse trees.

When this idea is used in a straightforward manner we get the following results:

	F_L	F_A	ULAS
simple “PCFG”	0.923	0.833	0.878

This approach corrects 38% of errors made by monkey dendrologist when counted only on labels and 31% counted on all attributes.

The PCFG model is rather simplistic as it takes into the account only labels of non-terminals and not complete sets of attributes. In the following we tried to enrich the information taken into the account by adding selected attributes.

The most obvious problem concerns arguments of verbs. The Świgr grammar analyses the sentence (zdanie) as a finite verbal phrase (ff) and a sequence of required phrases (arguments, fw) and free phrases (adjuncts, fl). For example, in Fig. 1 there are two zdanie nodes. The upper one consists of a required phrase realised by a nominal phrase in instrumental, a finite phrase and a required phrase representing the subject (nominal in nominative). The second zdanie comprises a subject (realised by a pronoun), finite phrase, required phrase realised by a prepositional-nominal complement and a free phrase representing prepositional-nominal adjunct. The required phrases (in particular subjects and complements) are indistinguishable for the pure PCFG algorithm.

In the first experiment the labels for required phrases were augmented with types of these phrases, e.g., subj, np(inst), infp (infinitival phrase), prepnp(‘z’,gen), and so on. Note that these symbols include in particular the value of case for required nominal and prepositional-nominal phrases.

We have also added several morphological features: gender, number and person (denoted GNP below). Note that since these attributes of nodes copy the features of the centre of the phrase, this provides the algorithm with data similar to that used with what is called “lexicalisation” in the context of PCFG (Collins 1997).

	F_L	F_A	ULAS
“PCFG”+fw-type	0.941	0.875	0.921
“PCFG”+GNP	0.936	0.876	0.915
“PCFG”+fw-type+GNP	0.932	0.873	0.914

Adding type of required phrases improves the results. This variant of the algorithm is able to avoid 46 % of errors made by a monkey dendrologist. Adding of gender-number-person improves results as well. A bit of surprise is that adding both elements results in slightly worse results than adding types alone. Probably in that case the training data gets too sparse. Note that with the added information various combinations of attributes are treated as completely independent non-terminals.

When the algorithm encounters a combination of children that was not seen in the training data, it uses a small smoothing value as a probability. We have counted the number of such unseen combinations in some variants of the experiment:

	Types	Occurences
simple “PCFG”	3,434	171,130
“PCFG”+fw-type	15,472	248,946
“PCFG”+fw-type+GNP	61,281	416,605

The growth of combinations with attributes added turns out to be very rapid, which unfortunately means that some kind of feature selection would be needed to train a manageable model. The vast majority of these combinations appear in realisations of the nominal phrase *fno* (where various kinds of attachments can happen at various levels) and in the sentence *zdanie* (where various combinations of complements and adjuncts are possible).

5 Experiments with Extended Version of Składnica treebank

Since the time the above experiments were conducted, the Składnica treebank has been extended by 2000 new annotated sentences. As the above research

showed that the treebank holds too sparse data, experiments on 25% bigger data set could be expected to give better results.

Baseline results, compared to these for the previous version of Składnica, are significantly worse:

	F_L	F_A
Mean monkey	0.806	0.676
Monkey	0.842	0.735

The ambiguity level of nodes, measured as the number of possible grammar productions that can be used to generate a given node, has grown by 10%. For comparison, we show a histogram (Fig. 2) presenting the percentage of nodes with a given number of possible productions (please note these are grouped in non-equal buckets). The value of 1 corresponds to unambiguous nodes and the

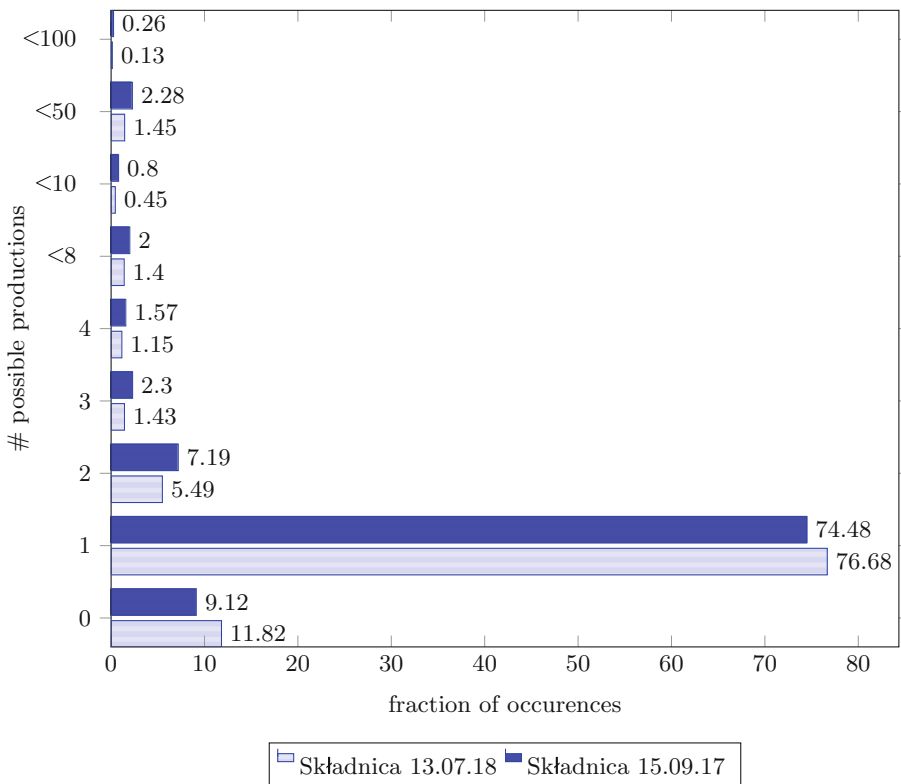


Fig. 2. Changes in the ambiguity level of nodes in two versions of Składnica

value of 0 corresponds to tree leaves. As can be seen in Fig. 1, the extended treebank has more nodes in each group of ambiguous nodes and less in unambiguous ones.

The following table lists results of experiments from the previous section repeated on the larger treebank. We do not provide the ULAS measure, since we do not have (yet) the dependency version of the present Składnica.

	F_L	F_A
simple “PCFG”	0.935	0.857
“simple”+fw-type	0.934	0.864
“PCFG”+GNP	0.930	0.866
“PCFG”+fw-type+GNP	0.927	0.864

The simple PCFG model shows an improvement. This approach now corrects 59% of errors made by a monkey dendrologist on labels (46% on all attributes). Even without taking into account that the baseline has been lowered, the overall accuracy of the model raised. This is a promising result which shows that we are able to obtain better results with the growth of the treebank.

Unfortunately, the richer models show slightly worse performance than for the older treebank. As in the prior experiments, it leads us to the conclusion of training data being too sparse. It is worth noting that the new treebank contains new types of constructions (in particular clauses with missing verbs), which we expect to be harder to learn.

6 Complements and Adjuncts

One of the hard problems in describing the syntactic structure of sentences is connected with the distinction between complements and adjuncts. The distinction is much argued about by linguists. It is well established in the tradition, but lacks a set of clear tests that would be agreed upon by a majority of researchers. Some researchers argue for dropping this distinction completely (Vater 1978; Przepiórkowski 1999).

Figure 3 shows some of the alternative variants of the inner sentence in Fig. 1, which differ in the pattern of complements and adjuncts. It is worth noting that all these structures are consistent with the valency frame for ‘to return’, which allows for the subject and an adjectival phrase (which gets realised here by a prepositional-nominal phrase).

After a discussion, annotators of the treebank decided that for the verb ‘to return’ the ‘to the country’ dependent is a complement but ‘from the U.S.’ is an adjunct. This decision seems to some extent arbitrary or at least based on deep semantics of the verb. The left tree of Fig. 3 shows that the parser can as well generate an interpretation where these two elements are interpreted the other way around. The right example shows a variant with only one complement being

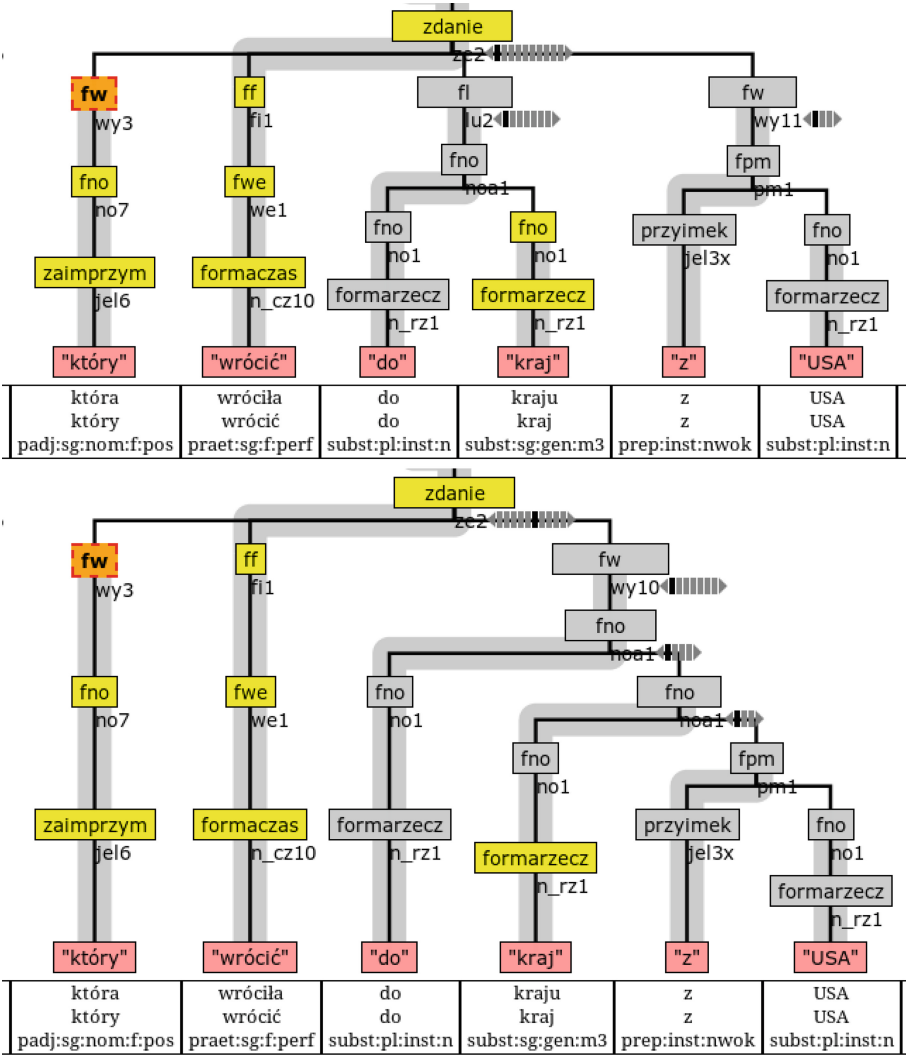


Fig. 3. Two of the other possible subtrees for the inner *zdanie* node from Fig. 1

a combined prepositional-nominal phrase which contains a sub-phrase ‘country from the U.S.’ which syntactically is perfectly acceptable (‘electronics from the U.S.’). If complements and adjuncts were not marked, the left tree of Fig. 3 would become identical to the tree in Fig. 1, leaving ambiguity only in real structural differences exemplified by the right tree.

The next of our experiments checks to what extent dropping the complement/adjunct distinction could help in disambiguating parse trees.

For that experiment we have modified the structure of *Składnica* by removing all nodes representing required and free phrases (*fw* and *fl*). These nodes have

just one child in the tree, so after the change the child takes the place previously occupied by the required or free phrase (compare Figs. 1 and 4).

The following table shows results of experiments repeated on such data:

	F_L	F_A	ULAS
Monkey	0.935	0.890	0.831
simple ‘PCFG’	0.960	0.922	0.890
‘PCFG’+GNPC	0.943	0.925	0.859

First of all it should be noted that the random baseline changes under such conditions. Strikingly, it gets better than simple PCFG-like algorithm on unchanged trees. ULAS does not change, but that is expected since the complement/adjunct distinction does not influence the shape of dependency trees (it would influence their labels).

The mostly visible change is in F_A for the simple PCFG-like algorithm. It gets better by almost 9% points when the complements/adjuncts distinction is ignored.

The third row of the table describes an experiment with labels augmented with gender, number, person, and case (which was included here because the case information from fw-type is no longer present). The addition of attributes improves a bit F_A but spoils F_L and ULAS, again probably due to sparseness of data.

These results suggest that indeed it may be reasonable to ignore the complement/adjunct dichotomy at the purely syntactic level. Perhaps the distinction could be reintroduced while considering semantics including semantic features of particular verbs.

We have also taken a closer look at decisions made by the algorithm at the level of *zdanie* (sentence). In the table below we show percentages of cases when the algorithm selects too few or too many constituents for *zdanie* compared to the gold standard.

	Too few	Too many
	Constituents	
‘PCFG’+fw-type	4.2%	15.0%
simple ‘PCFG’ no fw/fl	2.1%	26.3%

The data shows that the PCFG-like algorithm tends to choose productions that split sentences in a too granular way. Unfortunately the effect gets more pronounced when complement/adjunct distinction is ignored.

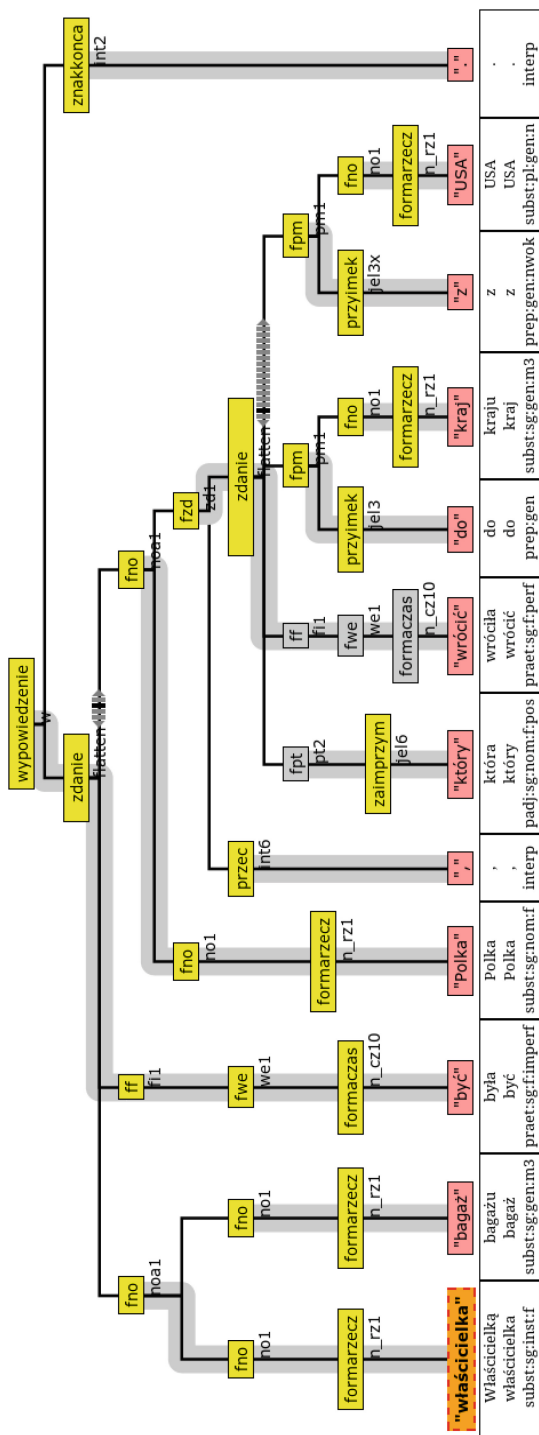


Fig. 4. The tree from Fig. 1 with all fw and fl nodes removed.

7 Summary and Outlook

In this paper we have explored a classical model of PCFG applied to the Polish data. The results are probably biased by the fact we use manually disambiguated morphological descriptions. They would probably be worse if a tagger was used. Nonetheless, we find the results better than we would expect from such a simple model.

In particular, the results are comparable to those of Wróblewska and Woliński (2012), who report 0.922 as ULAS of the best dependency parser trained on Składnica. It is worth noting that our algorithm selects among trees accepted by the non-probabilistic parser, so we have a guarantee that the selected structure is complete and in some way sound. This is hard to achieve in the case of probabilistic dependency parsers, which sometimes generate, e.g., a sentence with two subjects. On the other hand the present algorithm needs a parse forest as its input data, so it can produce trees only for sentences accepted by Świgrą. The probabilistic dependency parsers on the other hand produce some result for any sentence.

While the data presented here is already interesting, we have the feeling that we have only scratched the surface. In future experiments we intend to study the errors made by the algorithm. We will try to use extensions to PCFG that were proposed in the literature. But to incorporate selected attributes of nodes without causing the data to become too sparse it may be better to change the method to some form of regression based modelling.

References

- Abney, S., Flickenger, S., Gdaniec, C., Grishman, C., Harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J., Liberman, M., Marcus, M., Roukos, S., Santorini, B., Strzalkowski, T.: Procedure for quantitatively comparing the syntactic coverage of english grammars. In: Black, E. (ed.) *Proceedings of the Workshop on Speech and Natural Language, HLT 1991*. Association for Computational Linguistics, Stroudsburg (1991)
- Billot, S., Lang, B.: The structure of shared forests in ambiguous parsing. In: *Meeting of the Association for Computational Linguistics* (1989)
- Collins, M.: Three generative, lexicalised models for statistical parsing. In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, ACL 1998*. Association for Computational Linguistics, Stroudsburg (1997)
- Pereira, F., Warren, D.H.D.: Definite clause grammars for language analysis—a survey of the formalism and a comparison with augmented transition networks. *Artif. Intell.* **13**, 231–278 (1980)
- Przepiórkowski, A.: On complements and adjuncts in Polish. In: Borsley, R.D., Przepiórkowski, A. (eds.) *Slavic in HPSG*, pp. 183–210. CSLI Publications, Stanford (1999)
- Świdziński, M., Woliński, M.: Towards a bank of constituent parse trees for Polish. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) *TSD 2010*. LNCS, vol. 6231, pp. 197–204. Springer, Heidelberg (2010)

- Vater, H.: On the possibility of distinguishing between complements and adjuncts. In: Abraham, W. (ed.) *Valence, Semantic Case and Grammatical Relations*. Studies in Language Companion Series (SLCS), vol. 1, pp. 21–45. John Benjamins, Amsterdam (1978)
- Woliński, M., Głowińska, K., Świdziński, M.: A preliminary version of Składnica—a tree-bank of Polish. In: Vetulani, Z. (ed.) *Proceedings of the 5th Language & Technology Conference*, Poznań (2011)
- Wróblewska, A., Woliński, M.: Preliminary experiments in Polish dependency parsing. In: Bouvry, P., Kłopotek, M.A., Leprévost, F., Marciniak, M., Mykowiecka, A., Rybiński, H. (eds.) *SIIS 2011*. LNCS, vol. 7053, pp. 279–292. Springer, Heidelberg (2012)