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

Marcin Woliński, Dominika Rogozińska

Institute of Computer Science
Polish Academy of Sciences

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 Świgra 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 Świgra 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 Świgra 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 Świgra 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 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 percentage point). This is expected: 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 decomposi-

[1] `http://zil.ipipan.waw.pl/Składnica`

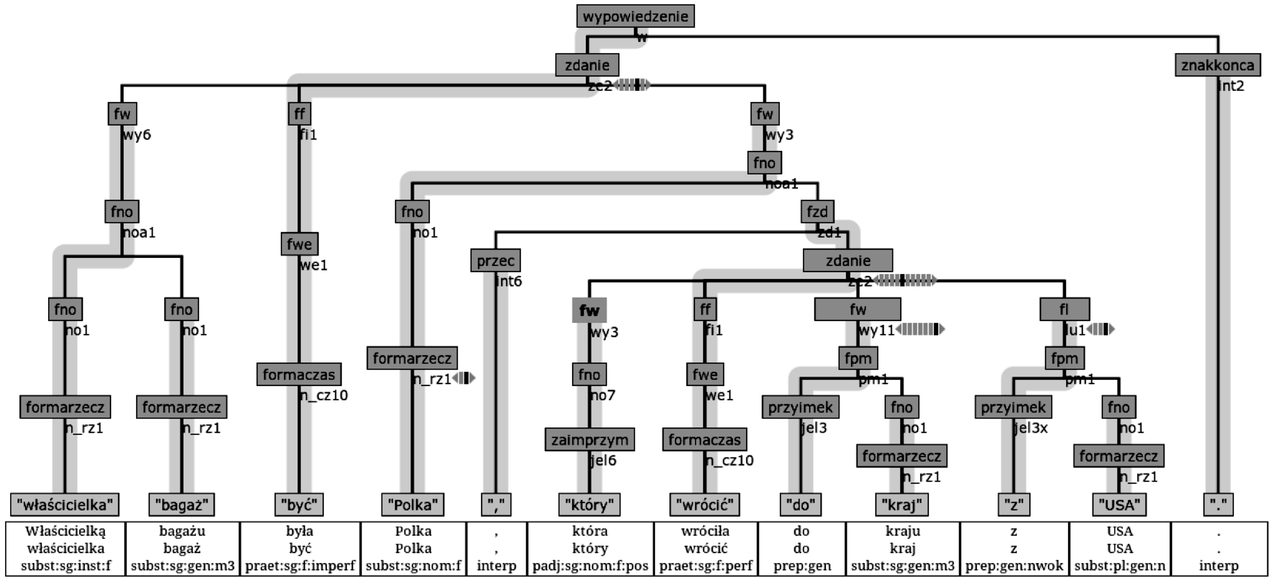

Figure 1: A Składnica tree for the sentence

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.'

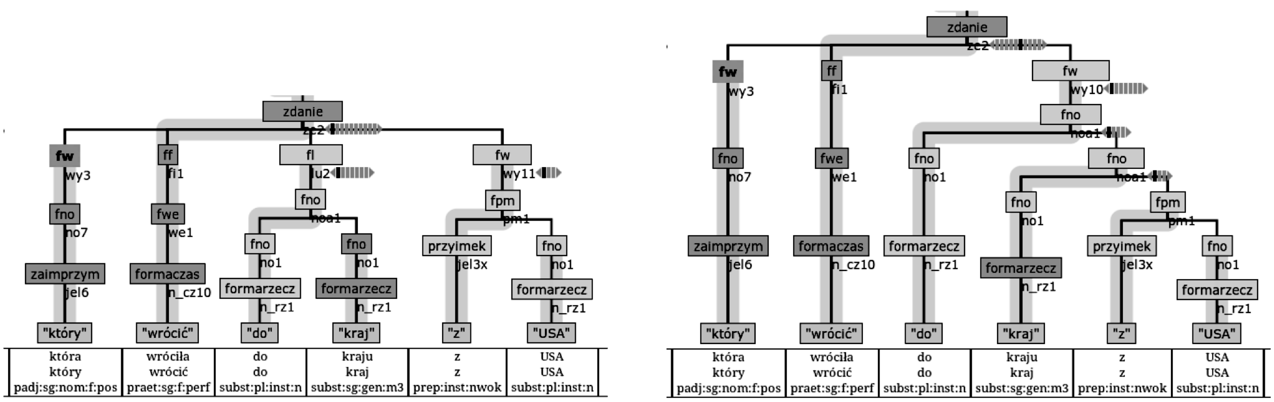

Figure 2: Two of the other possible subtrees for the inner zdanie node from Fig. 1

tion 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 Świgra 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.

Estimated probabilities of rules in a PCFG are 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.

The grammar of Świgra 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 Świgra 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. 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 2 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

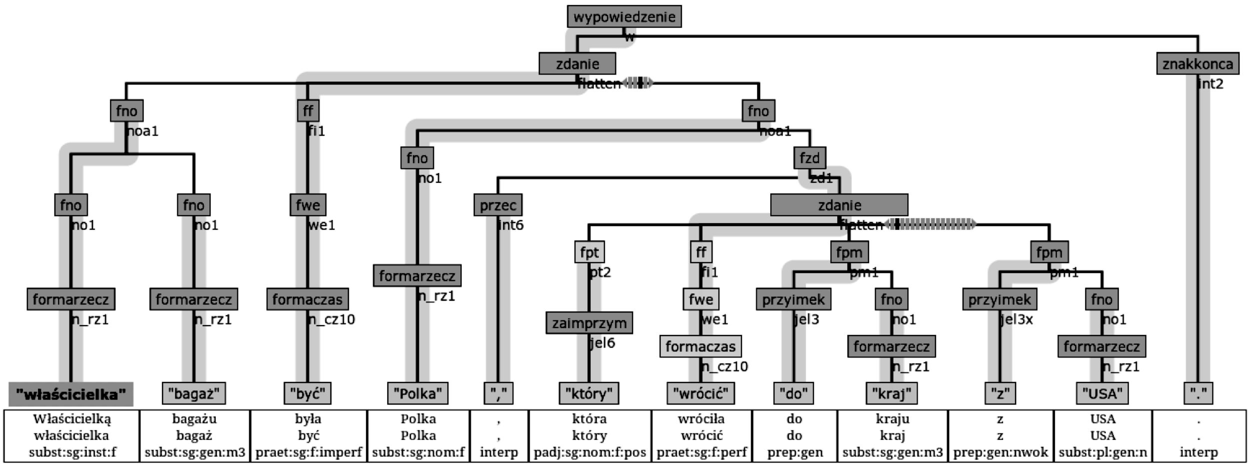

Figure 3: The tree from Fig. 1 with all fw and fl nodes removed.

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. 2 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 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. 2 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 Fig. 1 and 3).

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 percentage 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 algoritm 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.

6. 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 Świgra. 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.

7. References

Abney, S., S. Flickenger, C. Gdaniec, C. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski, 1991. Procedure for quantitatively comparing the syntactic coverage of english grammars. In E. Black (ed.), *Proceedings of the workshop on Speech and Natural Language*, HLT '91. Stroudsburg, PA, USA: Association for Computational Linguistics.

Billot, Sylvie and Bernard Lang, 1989. The structure of shared forests in ambiguous parsing. In *Meeting of the Association for Computational Linguistics*.

Collins, Michael, 1997. 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 '98. Stroudsburg, PA, USA: Association for Computational Linguistics.

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

Przepiórkowski, Adam, 1999. On complements and adjuncts in Polish. In Robert D. Borsley and Adam Przepiórkowski (eds.), *Slavic in HPSG*. Stanford: CSLI Publications, pages 183–210.

Świdziński, Marek and Marcin Woliński, 2010. Towards a bank of constituent parse trees for Polish. In Petr Sojka (ed.), *Text, Speech and Dialogue, 13th International Conference, TSD 2010, Brno, September 2010, Proceedings*, volume 6231 of *LNAI*. Heidelberg: Springer.

Vater, Heinz, 1978. On the possibility of distinguishing between complements and adjuncts. In Werner Abraham (ed.), *Valence, Semantic Case and Grammatical Relations*, volume 1 of *Studies in Language Companion Series (SLCS)*. Amsterdam: John Benjamins, pages 21–45.

Woliński, Marcin, Katarzyna Głowińska, and Marek Świdziński, 2011. A preliminary version of Składnica—a treebank of Polish. In Zygmunt Vetulani (ed.), *Proceedings of the 5th Language & Technology Conference*. Poznań.

Wróblewska, Alina and Marcin Woliński, 2012. Preliminary experiments in Polish dependency parsing. In P. Bouvry, M.A. Kłopotek, F. Leprevost, M. Marciniak, A. Mykowiecka, and H. Rybinski (eds.), *Security and Intelligent Information Systems. International Joint Confererence, SIIS 2011, Warsaw, Poland, June 13-14, 2011, Revised Selected Papers*, volume 7053 of *LNCS*. Springer.