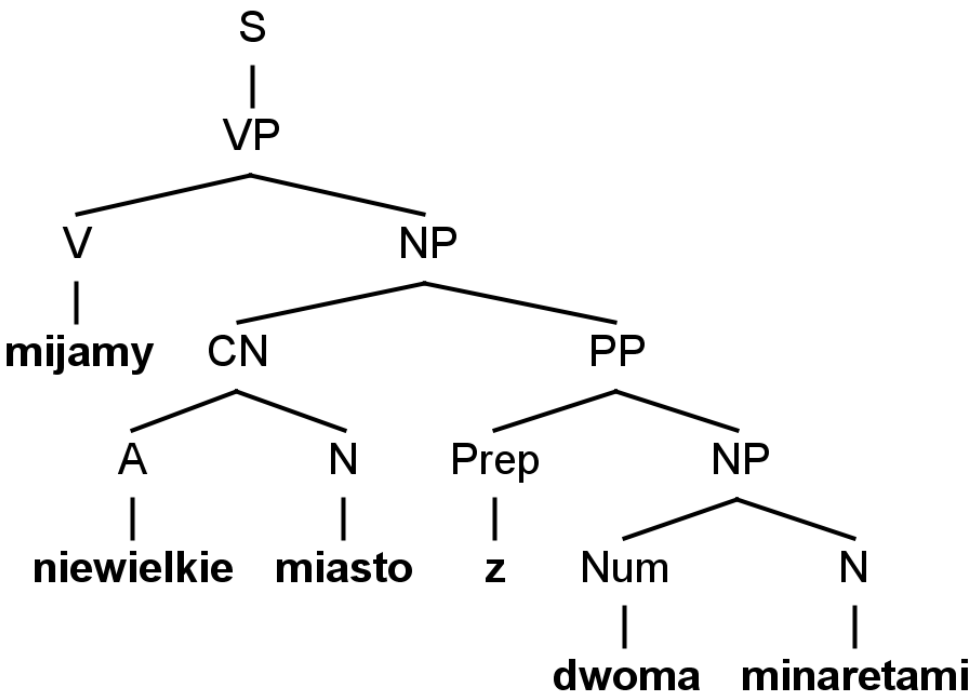
Maria Irena Szawerna

Lab 1: Chapter 4 report

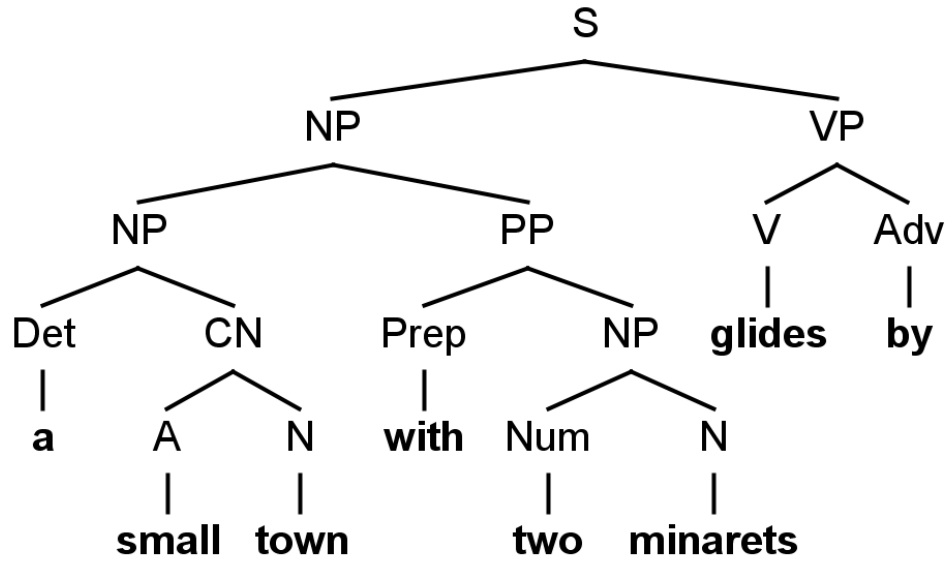
1. Phrase structure trees

Having consulted Arianna, I decided to construct phrase structure trees for four sentences in both languages (a total of 8 trees). While, naturally, I will need more practice with this format, I already have some experience with it, and having more trees could have proven to be a lot to compare in the upcoming stages. The recommendation I was given was to do between 6 and 10 trees, and that is what I did. For drawing the trees I used software called TreeForm. It does not parse the sentences for me, it only allows me to draw and edit the trees rather easily, and I used it in another class on syntax a few years ago. I have also received permission in this class to use it. I decided to disregard sentence-final punctuation in my examples, as per the examples in the presentation, but if needed, I can edit it in (and from the gfud trees I can see that for that I need to have an *Utt* node with another *Utt -> S* and *Punct* nodes connected to it). Some decisions I made while making the trees will be discussed when they are compared to the gfud generated trees.

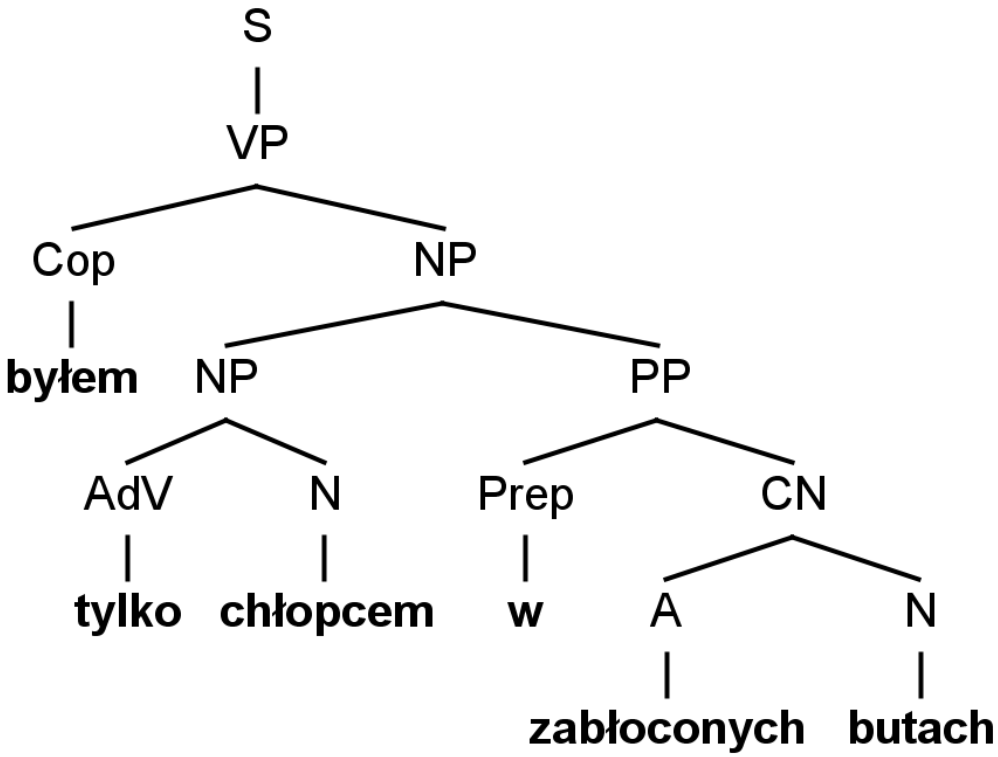
Tree 1:



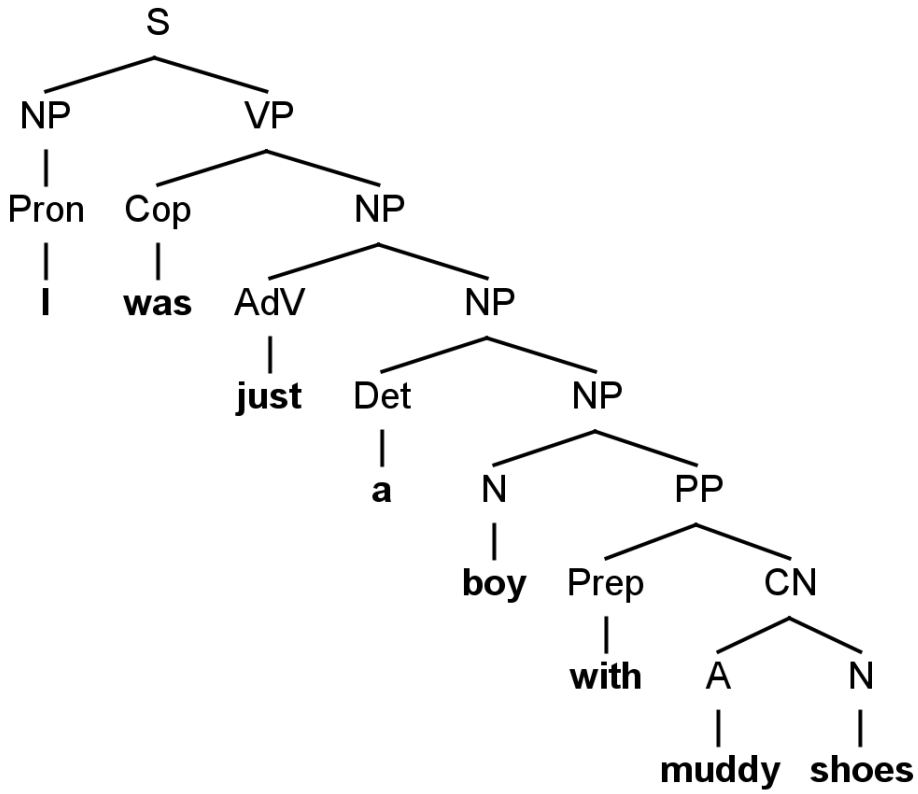
Tree 2:


S
NP
VP
NP
PP
V
Adv
Det
CN
Prep
NP
glides
by
a
A
N
with
Num
N
small
town
two
minarets

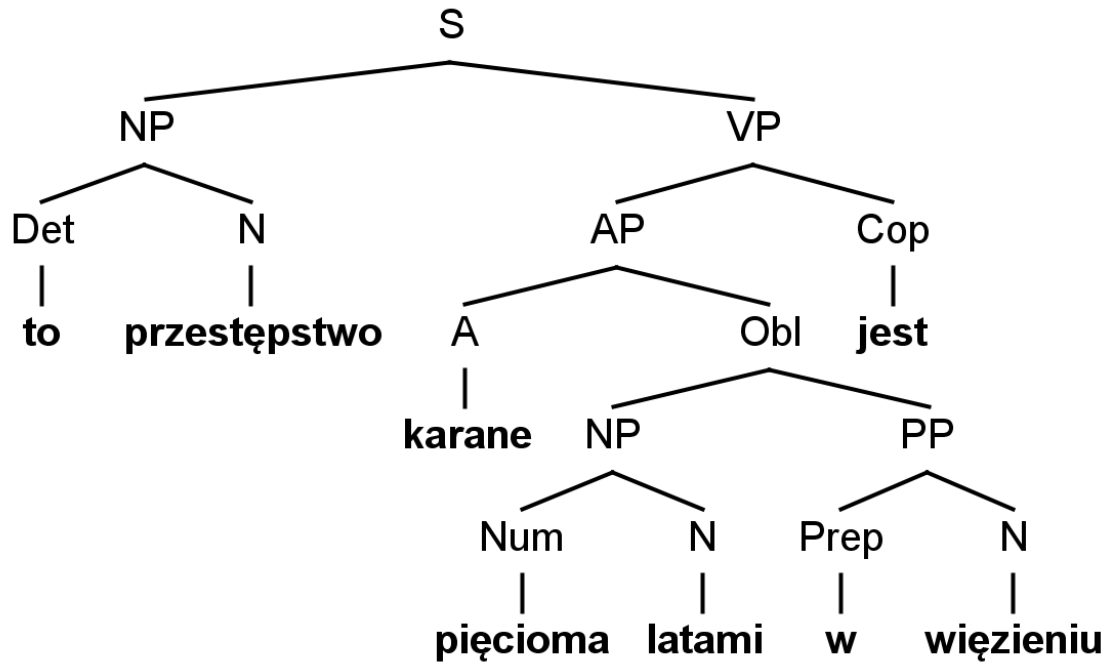
Tree 3:


S
VP
Cop
NP
byłem
NP
PP
AdV
N
Prep
CN
tylko
chłopcem
w
A
N
zabłoconych
butach

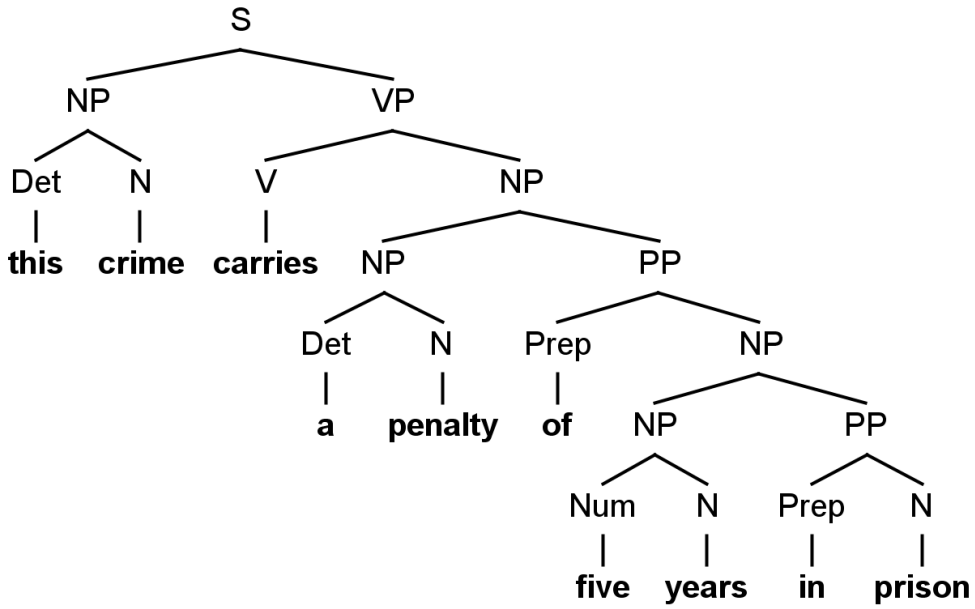
Tree 4:


S
NP
VP
Pron
Cop
NP
I
was
AdV
NP
just
Det
NP
a
N
PP
boy
Prep
CN
with
A
N
muddy
shoes

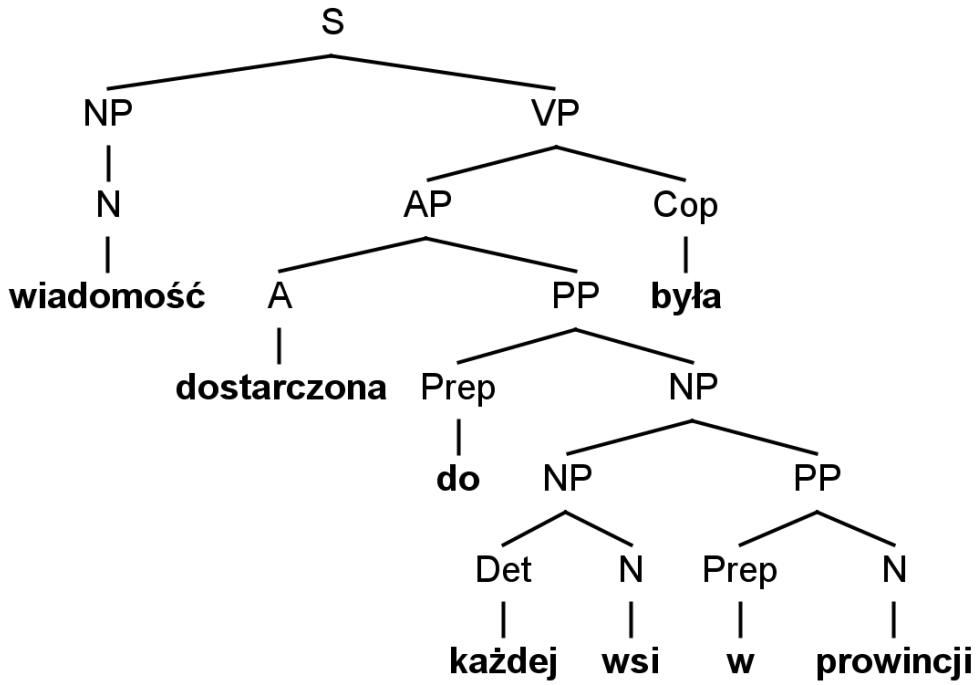
Tree 5:


S
NP
VP
Det
N
AP
Cop
to
przestępstwo
A
Obl
jest
karane
NP
PP
Num
N
Prep
N
pięcioma
latami
w
więzieniu

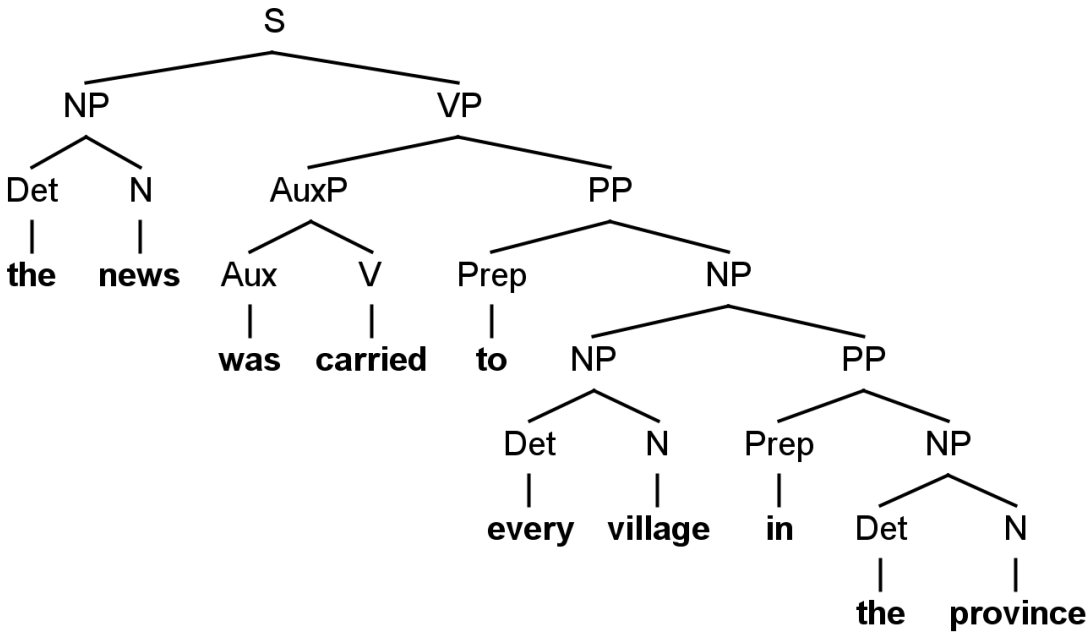
Tree 6:


S
NP
VP
Det
N
V
NP
this
crime
carries
NP
PP
Det
N
Prep
NP
a
penalty
of
NP
PP
Num
N
Prep
N
five
years
in
prison

Tree 7:


S
NP
VP
N
AP
Cop
wiadomość
A
PP
była
dostarczona
Prep
NP
do
NP
PP
Det
N
Prep
N
każdej
wsi
w
prowincji

Tree 8:



2. Test the grammar

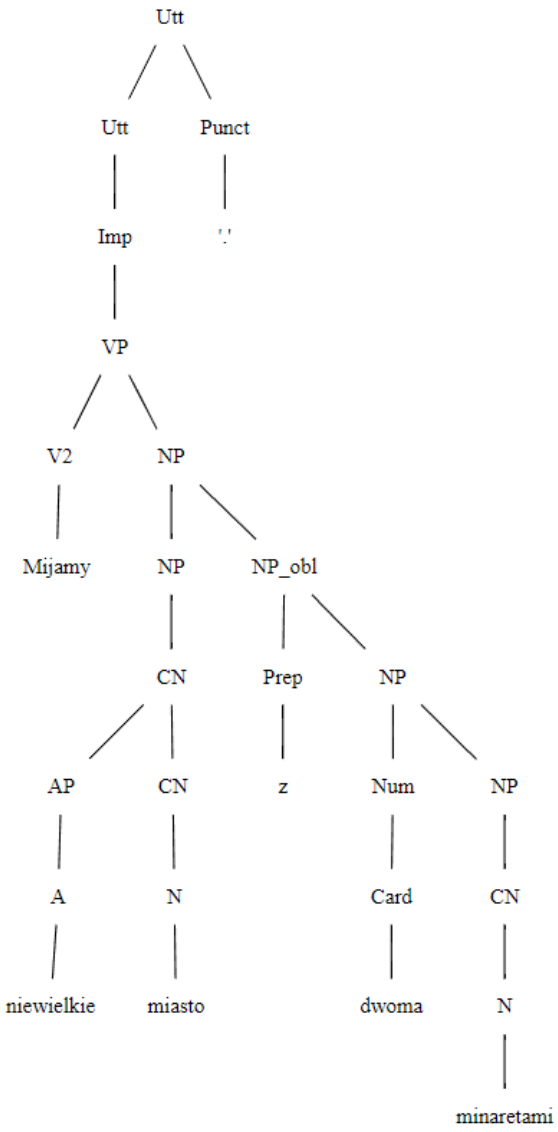
For testing the grammar quantitatively, I first tries to use non-POS tagged sentences, which turned out to be a disaster, with hardly anything properly marked and attached, as the grammar does not include a lexicon that would account for most of the vocabulary in the sentences. Instead, I used the files that had the POS tags (which I luckily annotated in the text files with the translations, before transforming them to conllu files in the previous week's assignment). All of the gfud elements were done on eduserv, as I am unable to get a working version of the tool locally; I also had to use the solution that I previously shared in the program Discord so as not to get errors caused by encoding.

I first used gfud to create a conllu file containing the parses based on English.dbnf. This meant running the following command: *cat comp-syntax-corpus-english.txt | gfud dbnf English.dbnf Utt > my_english.conllu* and then evaluating the created file against my hand-annotated conllu file from last week: *gfud eval macro LAS english.conllu my_english.conllu*. I obtained the following results: ***UDScore {udScore = 0.6614094842355712, udMatching = 20, udTotalLength = 289, udSamesLength = 185, udPerfectMatch = 1}***, which seems to be quite good, granted that the same grammar only got a udScore of ~0.5 when tested in the lecture. Naturally, the grammar must be lacking some of the structures that appear in the corpus (which contains some really complex sentences), and it is also possible that my annotation is faulty at times (meaning that the grammar analyzed the sentence better than I did, but the results do not match up and my annotation is treated as the golden standard regardless of its quality).

I then continued to do the same for Polish: the problem here though was that gfud refused to read my conllu file with any sorts of comments in it, stating *gfud: ERROR: # sent_id = 1 incomplete UDWord*. I solved that by creating an additional file, polish-02.conllu, which had the same contents but seemed not to have the thing that caused errors. Thus, the commands I ran looked like this: first *cat comp-syntax-corpus-polish.txt | gfud dbnf English.dbnf Utt > my_polish.conllu*, and then *gfud eval macro LAS polish_02.conllu my_polish.conllu*. I obtained the following evaluation: ***UDScore {udScore = 0.4356202620908503, udMatching = 17, udTotalLength = 186, udSamesLength = 78, udPerfectMatch = 1}***. This is a significantly worse score than the one obtained for English. While the same may hold true as for the English evaluation (that my mistakes in the annotation may influence it), the magnitude of the difference indicates that the grammar rules that work for English do not work all that well for Polish. The reason behind it may be that Polish has a much more flexible word order; while sentences usually follow the SVO order, inversion can be used for emphasis and subject can be dropped. Adjectives can also modify nouns both by preceding and following them, and auxiliaries can follow verbs instead of preceding them as well; there are more differences, as per my submission for Chapter 2, but these I remember to occur in the corpora.

For qualitative testing I used the trees I constructed manually in step 1 and trees generated by gfud on eduserv. Since it was not possible for me to generate .pdf or proper LaTeX files on the server and the gfud tool does not work on my own machine, I had to use an online solution suggested by Arianna on Discord: using the tool found under this link to generate the tree from a .dot file generated by a command following this format: *echo 'Wiadomość:<NOUN> dostarczona:<ADJ> była:<AUX> do:<ADP> każdej:<DET> wsi:<NOUN> w:<ADP> prowincji:<NOUN> .:<PUNCT>' | gfud dbnf English.dbnf Utt | gf-ud parse2latex myparsetree* – so first echoing the POS-annotated sentence, using English.dbnf to parse it, and then asking gfud to generate a tree out of that. This procedure worked for all the English sentences, and one Polish sentence out of the ones I made trees for (it worked on some other ones as well). The problem here is that the tool claims that it "gf-ud: cannot parse abstree". I assume this has something to do with the weird syntax of Polish and English.dbnf not being the best at parsing them (even the one tree that worked has some wrong labels). Thus, my capacity to make proper comparisons is a bit limited, but I will also try to use the bracket representation, not only the tree one, to do this comparison. In the following section the generated trees for each sentence will be presented and annotated with a comparison to the ones in part 1.

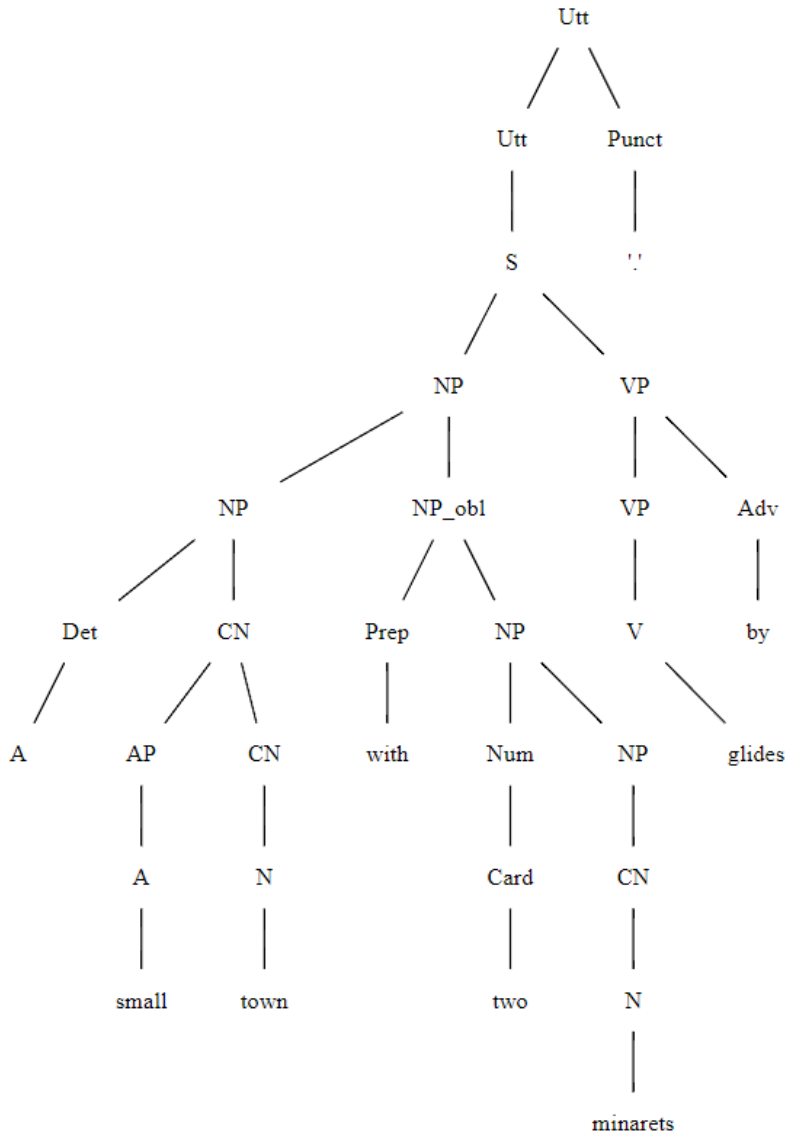
Tree 1:



This tree – the only one that was successfully generated for Polish – is actually quite accurate. The first thing that is worth noting is that gfud included the *Utt* node and punctuation (which I commented on in part 1). The one major difference is that it is classified as an imperative sentence, which is incorrect, as it is just a regular declarative sentence. This is because in English the only situation where a pronoun can be dropped is in imperative sentences, while in Polish pronoun-dropping is very common, since the information about person and number is conveyed by the inflected verb. Gfud also used NP_obl where I used PP (prepositional phrase). In addition, it always includes phrase names like AP, CN or category names like Card where I omitted them. I did not think to mark Card, and for the rest I went with having a NP potentially consist of just a N. Understandably, this is not what the rules used in

generating this tree claim, and this is where that difference comes from, but if one counts my omissions as an “abbreviation” of the tree, then it is really similar.

Tree 2:



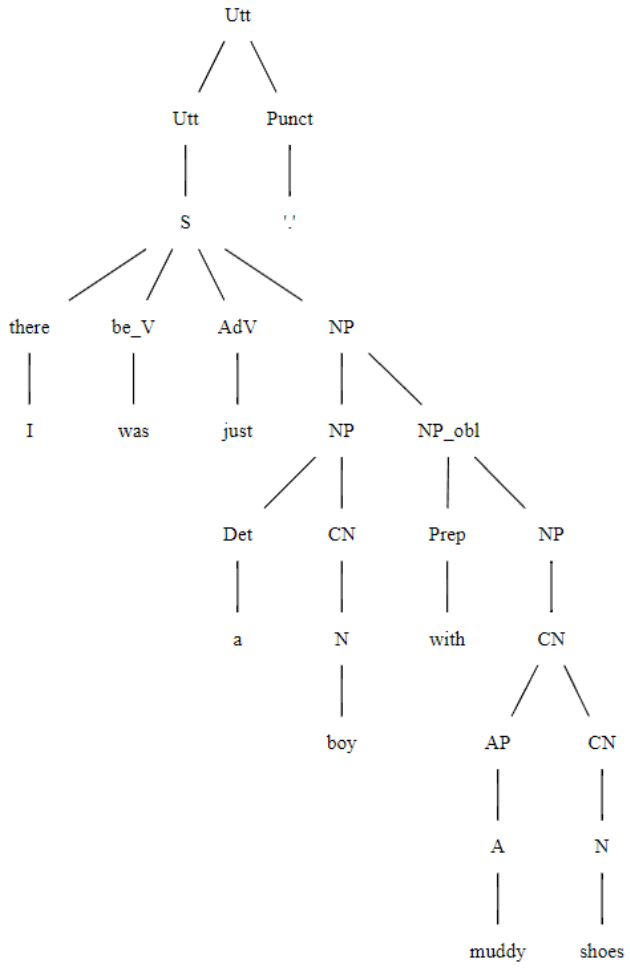
Similarly to the previous tree, this one aligns with mine, with the exception of the more detailed nodes.

Tree 3:

```
[gusszawma@GU.GU.SE@eduserv ~]$  echo 'Byłem:<AUX> tylko:<PART> chłopcem:<NOUN> w:<ADP> zabłoconych:<ADJ> butach:<NOUN>
.:<PUNCT>' | gfud dbnf English.dbnf Utt | gf-ud parse2latex myparsetree
gf-ud: cannot parse abstree (Chunks (Imp (VP (aux Byłem))) ('s' tylko) (Utt (Utt (NP (NP (CN (N chłopcem))) (NP_obl (Pre
p w) (NP (CN (AP (A zabłoconych)) (CN (N butach)))))))) (Punct '.')))
CallStack (from HasCallStack):
  error, called at GFConcepts.hs:22:8 in main:GFConcepts
```

This is the first of the three Polish trees that could not be generated. Seeing how even in the first one the type of the sentence was misclassified, I assume that this is because of the grammar not being suited for Polish. From the bracketed notation I can see that once again this sentence was classified as an imperative one, for the same reason as in Tree 1. I can also see that "tylko" – "just" in English – is classified as some really weird *'s'* tag. It, along with the copula (mislabelled here as *Aux*) is completely detached from the rest of the sentence, where *Utt* starts. It seems to me that that part is analyzed the same way as in my tree. It is likely all those issues that made it impossible for the tree to be generated.

Tree 4:



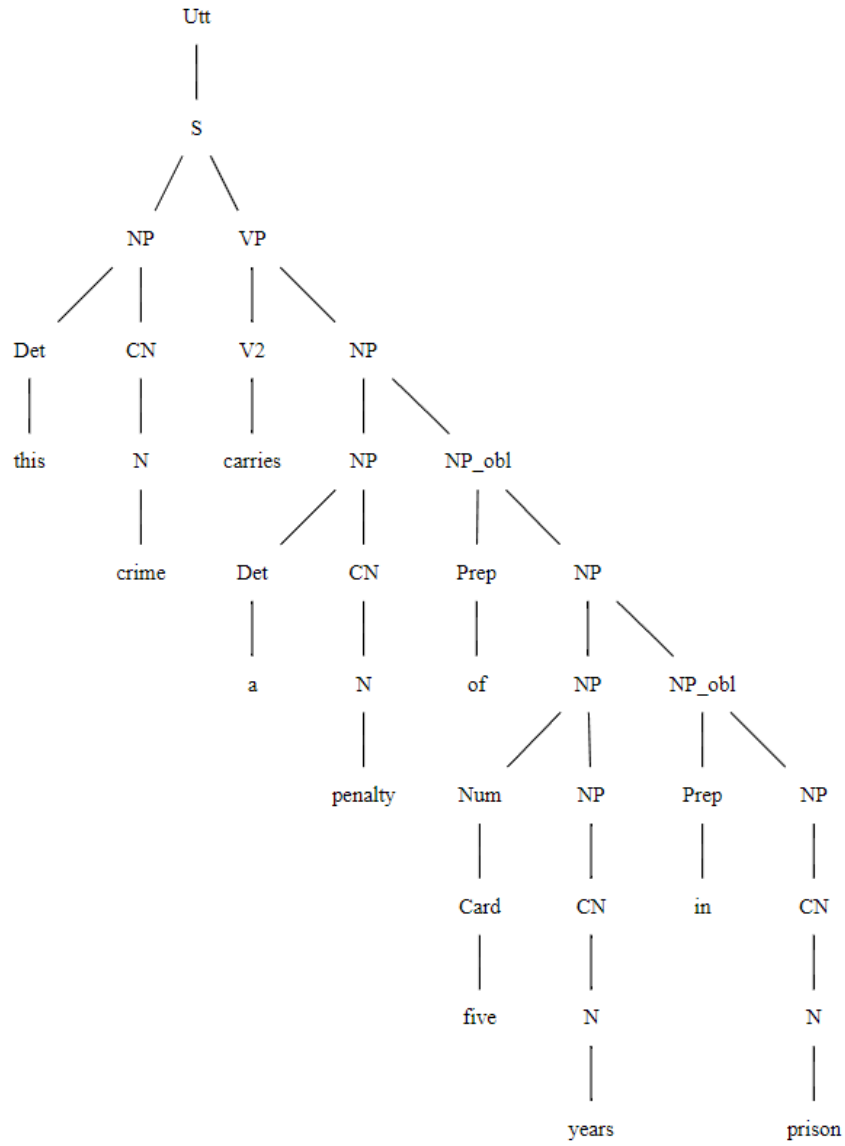
There is a number of differences between this tree and my analysis. First of all, in the gfud tree, four different nodes stem from *S*. This is what I was taught is not permitted in syntax trees, and this is why I attached those elements elsewhere (in a *VP* with the copula, following the lecture slides, and with *AdV* modifying a noun phrase). Another difference can be found in where the determiner "a" is attached: in the generated tree a separate *NP* splits from the *PP* (or NP_*obl*) that is divided into "a" and "boy", while for me "a" describes the entirety of "boy with muddy shoes," and I am uncertain which option is more correct.

Tree 5:

```
[gusszawma@GU.GU.SE@eduserv ~]$  echo 'To:<DET> przestępstwo:<NOUN> karane:<ADJ> jest:<AUX> pięcioma:<NUM> latami:<NOUN
> w:<ADP> więzieniu:<NOUN> .:<PUNCT>' | gfud dbnf English.dbnf Utt | gf-ud parse2latex myparsetree
gf-ud: cannot parse abstree (Utt (Utt (S (NP (Det To) (CN (CN (N przestępstwo)) (AP (A karane)))) (cop jest) (Comp (NP
(NP (Num (Card pięcioma)) (NP (CN (N latami)))) (NP_obl (Prep w) (NP (CN (N więzieniu)))))))) (Punct '.'))
CallStack (from HasCallStack):
  error, called at GFConcepts.hs:22:8 in main:GFConcepts
```

For this tree it is not that easy to identify for me what the issue with generating a visual version was. I guessed it is the inversion of object and verb (essentially having the SOV order in this sentence), but changing that did not help. At a second glance I think that this grammar has issues dealing with Polish copula verbs and how they connect to the other elements of the sentence. There is also an issue where in this analysis what I marked as *Obl* (it is not a PP since there is no preposition, but in English it would use one; the preposition is missing here because the declension (instrumental case) makes it obsolete) is classified as a *Comp* of the copula, I assume. This is incorrect; it is supposed to modify the actual complement, which here is classified as an adjective describing the subject. This is due to the SOV order, but it also messes up the generated tree a lot. The rest of the elements seem okay, again, with slightly more detailed nodes. It is perhaps worth mentioning that Polish is generally an SVO language, but inversion happens often for emphasis or stylistic purposes and does not impact the understanding of the language at all, since the bulk of the information that in English is contained in the word order is conveyed by inflection in Polish. Some sentences, when said in the "correct" SVO order sound wrong to natives.

Tree 6:



This tree seems to be almost identical to mine, with the exception of the aforementioned more detailed nodes.

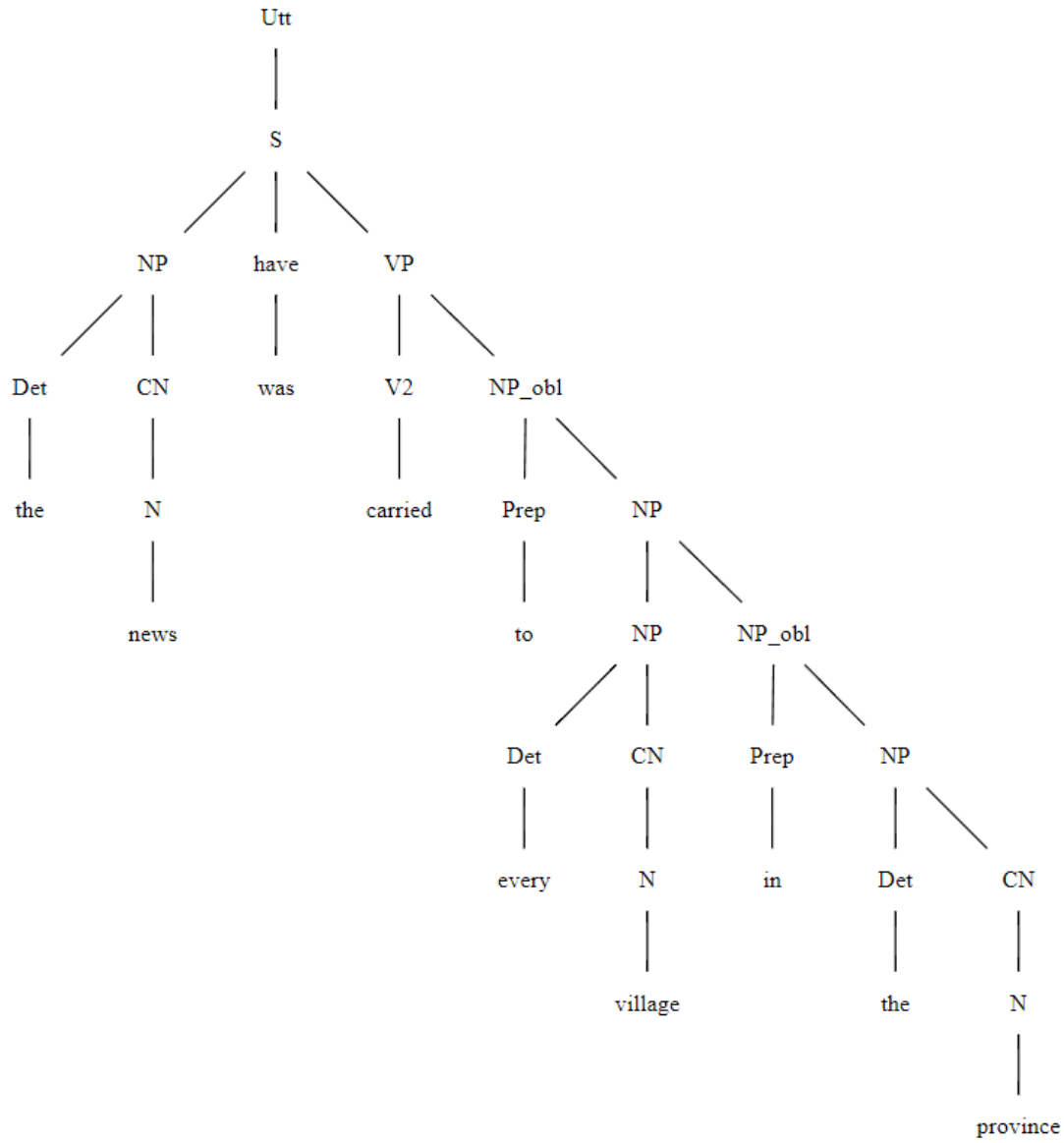
Tree 7:

```
[gusszawma@GU.GU.SE@eduserv ~]$  echo 'Wiadomość:<NOUN> dostarczona:<ADJ> była:<AUX> do:<ADP> każdej:<DET> wsi:<NOUN> w
:<ADP> prowincji:<NOUN> .:<PUNCT>' | gfud dbnf English.dbnf Utt | gf-ud parse2latex myparsetree
gf-ud: cannot parse abstree (Utt (Utt (S (NP (CN (CN (N Wiadomość)) (AP (A dostarczona)))) (cop była) (Comp (NP_obl (Pr
ep do) (NP (NP (Det każdej) (CN (N wsi))) (NP_obl (Prep w) (NP (CN (N prowincji))))))))) (Punct '.'))
CallStack (from HasCallStack):
  error, called at GFConcepts.hs:22:8 in main:GFConcepts
```

Similarly to tree 5, the inversed word order is problematic here. When reversed though, it still does not fix the issue of the tree not being printed out (though the complement is then properly identified). As for this one, again, the actual complement is mislabeled as a part of the subject noun phrase, and the *PP* describing that complement is misidentified as the

complement. I am also unsure if both of the prepositional phrases are properly attached, and it is hard to tell in this notation; I am not sure they are properly attached in my interpretation of the sentence either.

Tree 8:



There are a few differences between this tree and my tree, but the general structure is similar. Gfud again splits a node in three where I put a VP splitting into an AuxP and a PP. I guess the grammar does not have rules that would account for this passive construction here. Aside from that it is just a few more more detailed labels that set this tree apart from mine.

Overall the grammar seems to be doing a great job when it comes to English sentences, but it struggles a lot with Polish ones since the rules of the two languages differ so much.

3. Modify the grammar, test on UD treebanks

My first step for this task was to inspect the structure of the grammar. I then proceeded to edit the probably least important part, the lexicon, and test it a little bit with simple sentences. This proved to be challenging since for some reason I could not easily get the file to save in ISO-8 and instead it saved in some weird ASCII format that made it impossible for gfud to read. Another issue was the sheer variety of inflectional or declensional forms that Polish words can take, which meant that I invariably reduced the variety of words in the lexicon and likely omitted some forms since I was doing it from memory. I also had to omit our special letters in the words in the lexicon for it to work. A few tests can be seen below:

```
[gusszawma@GU.GU.SE@eduserv ~]$ echo 'Mary widzi piwo' | gfud dbnf Polish3.dbnf Utt

# text = Mary widzi piwo
# analyses = 2
# parsetree = (Utt (S (NP (PN Mary)) (VP (V2 widzi) (NP (CN (N piwo))))))
# weight = 1.953125e-3
1       Mary    _       PROPN   _       _       2       nsubj   _       _
2       widzi   _       VERB    _       _       0       root    _       _
3       piwo    _       NOUN    _       _       2       obj     _       _


[gusszawma@GU.GU.SE@eduserv ~]$ echo 'Mary widzi dobre piwo' | gfud dbnf Polish3.dbnf Utt

# text = Mary widzi dobre piwo
# analyses = 2
# parsetree = (Utt (S (NP (PN Mary)) (VP (V2 widzi) (NP (CN (AP (A dobre)) (CN (N piwo)))))))
# weight = 2.44140625e-4
1       Mary    _       PROPN   _       _       2       nsubj   _       _
2       widzi   _       VERB    _       _       0       root    _       _
3       dobre   _       ADJ     _       _       4       amod    _       _
4       piwo    _       NOUN    _       _       2       obj     _       _


[gusszawma@GU.GU.SE@eduserv ~]$ echo 'dziewczyna wczoraj spiewala' | gfud dbnf Polish3.dbnf Utt

# text = dziewczyna wczoraj spiewala
# analyses = 2
# parsetree = (Chunks (Comp (Comp (NP (CN (N dziewczyna)))) (Adv wczoraj)) (Imp (VP (V spiewala))))
# weight = 1.953125e-7
1       dziewczyna      _       NOUN    _       _       0       root    _       _
2       wczoraj _       ADV     _       _       1       advmod  _       _
3       spiewala        _       VERB    _       _       1       dep     _       _


[gusszawma@GU.GU.SE@eduserv ~]$ echo 'dziewczyny kochaja koty' | gfud dbnf Polish3.dbnf Utt

# text = dziewczyny kochaja koty
# analyses = 2
# parsetree = (Utt (S (NP (CN (N dziewczyny))) (VP (V2 kochaja) (NP (CN (N koty))))))
# weight = 9.765625e-4
1       dziewczyny      _       NOUN    _       _       2       nsubj   _       _
2       kochaja _       VERB    _       _       0       root    _       _
3       koty    _       NOUN    _       _       2       obj     _       _
```

It seems that all these simple sentences are analyzed properly.

I then proceeded to see what Polish treebanks there are that I can test my changes to the grammar on. There are three: UD_Polish-LFG, UD_Polish-PDB, and the one I already have used, UD_Polish-PUD. For the last one there is only a test version, but for the other two there

are train versions; however, those have proven to be too big to do any sort of testing on them, so I just used the test versions for all. I uploaded them to eduserv and used a part of a pipe from the slides: *cat pl_lfg-ud-test.conllu | gfud extract-pos-words > pl_lfg-sents.txt* (and analogous ones for the other .conllu files). This ended me up with .txt files with just the POS-annotated sentences, similar to the file we got to work on in the previous lab. I can now use these to try to get some result while testing the changes to the grammar.

I first tested what the udScores for those sentences with the English grammar were:

- For the PUD: UDScore {udScore = 0.3173780138949314, udMatching = 1000, udTotalLength = 18438, udSamesLength = 5650, udPerfectMatch = 7}
- For the LFG: UDScore {udScore = 0.4830695754248489, udMatching = 384, udTotalLength = 3058, udSamesLength = 1364, udPerfectMatch = 48}
- For the PDB: UDScore {udScore = 0.35819580605629137, udMatching = 2215, udTotalLength = 33867, udSamesLength = 10598, udPerfectMatch = 77}

There is quite a difference between the lower end (PUD – 32%, PDB – 36%) and the upper end (LFG – 48%). I decided to keep testing my changes on PUD, and once I would get a satisfactory performance increase, I would compare how that affected the other two.

I then proceeded to edit the grammar itself. Since there are many rules in it, after first trying to go through them one by one, I decided to move on to changing the most general ones, or the ones that would likely have impact on a lot of parses. I deleted a number of rules that made use of the extensive set of English auxiliaries (e.g. do-support), which were irrelevant for Polish. Then I started by making it possible for a sentence to have no overt subject (to counteract the issues from part 2). These initial changes only improved the performance by one percent point, but it meant that I am heading in the right direction. To my surprise, my next few changes (including the auxiliary in a post-verbal position) actually decreased the score, which I do not entirely understand. Adding a rule that allowed for subject/verb inversion for basic sentences increased the performance, so I followed that path; expanding that to a few more sentence structures was reflected in a slight performance increase. Trying to extend that to object/verb inversion was not nearly as successful. The problem here is that the grammar does not capture any of the inflectional or declensional endings, and I highly doubt that I can significantly improve its performance without that being included, but implementing all of that would be a titanic effort way beyond this assignment in my opinion. It would be much easier to improve the performance of a grammar like this for a language whose structure is similar to English, like, say, Swedish. I removed all of those changes as they impacted the performance in a very negative way (decreasing it by about 1 percent point). The same issue, but much more hurtful

to the score, occurred when I tried to add an option for oblique phrases to not require a preposition (since in Polish many of those can be reflected using case endings, such as the instrumental or the locative). This, however, caused the score to drop by a good few percent points since, again, it could not tell an *obl* from an *obj*, for example; this is something for which incorporating cases would have made a huge difference, but I am also not sure how the grammar would "detect" those if they are not included in the POS tags to begin with. Another small improvement I made was allowing for *CN* phrases to have the *AP* come either before or after the *N*, which is definitely possible in Polish. I got quite an increase by also allowing the same "free" order for *Num* and *Sym* in a *NP*. I made it possible for an *xcomp* after a *VV* to not be to-marked, since it is not how we mark infinitives in Polish. At this point I think I have reached the upper bound of what I can alter without having explicit case markings. The grammar does not account for possessives other than possessive pronouns, since it cannot tell nouns in genitive from other nouns. It does not allow for a free word order for ditransitive verbs since it cannot tell apart objects in dative and accusative. For the same reason other inversions decreased the score too, since telling nominative from accusative is also not possible. Since, as shown in parts 1 and 2, Polish does make use of inversion, this brings down the score quite a bit. I decided to test how this grammar performs now on all three of the treebanks:

- For the PUD: UDScore {udScore = 0.33056011579050615, udMatching = 1000, udTotalLength = 18438, udSamesLength = 5868, udPerfectMatch = 7}
- For the LFG: UDScore {udScore = 0.5493675688486042, udMatching = 384, udTotalLength = 3058, udSamesLength = 1572, udPerfectMatch = 45}
- For the PDB: UDScore {udScore = 0.3760957633461008, udMatching = 2215, udTotalLength = 33867, udSamesLength = 11058, udPerfectMatch = 75}

Aside from the LFG treebank, where the improvement was by ~6 percent points, the rest of the improvement was not so stunning, by only about 2 pp. However, I listed the reasons for that above. I am also curious as to why the LFG treebank is so much "better" at this task than the other ones. I assume it is composed of simpler sentences that follow the regular word order. I hope that not reaching 60% score in this task was okay. I have talked with Aarne and he admitted that it is indeed difficult to represent non-analytic languages in this format, so the performance is somewhat justified. The Polish grammar I ended up with can be found here.