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](https://sourceforge.net/projects/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:

Diagram

Description automatically generated

Tree 2:

Diagram, schematic

Description automatically generated

Tree 3:

Diagram

Description automatically generated

Tree 4:

Diagram

Description automatically generated

Tree 5:

Diagram

Description automatically generated

Tree 6:

Diagram

Description automatically generated

Tree 7:

Diagram

Description automatically generated

Tree 8:

Diagram, schematic

Description automatically generated

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](https://dreampuf.github.io/GraphvizOnline) 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:

A picture containing diagram

Description automatically generated

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.

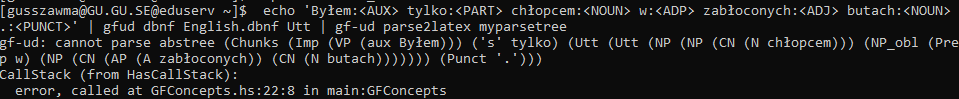
Tree 2:

Diagram

Description automatically generated

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

Tree 3:



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.

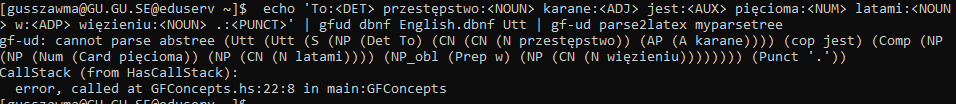
Tree 4:

Diagram

Description automatically generated

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:



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.

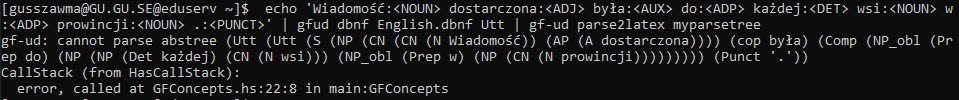
Tree 6:

Engineering drawing

Description automatically generated

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

Tree 7:



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:

A picture containing chart

Description automatically generated

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