Sergei Chestakov
913914694

Natural Language Processing (ECS 189g) HW 3

Task 1
Based on the grammar and lexicon files, this PCFG implements a unigram model because all of the grammar and lexicon rules are weighted equally. Since everything has an equal weight, the probability of a parse given a sentence is essentially random given the choices, especially since any POS can follow any other one. This means that any word in the vocabulary can come next, regardless of the context

Task 2
The majority of the sentences parsed with the PCFG created from just grammar1 result in failures due to the lack of grammar rules defined in the grammar1 file (8 compared to grammar2's 50). This leads to an inflexibility in tree structure which often results in sentences that cannot be parsed into a complete tree. When the two are merged, however, it works much better, since grammar2's rules were already sufficient for all of the test sentences and grammar1 just adds some more structure to it.

Task 3
Grammar1 has very few rules, but they have varying weights, which results in a lot of similar sentences which are almost all 4-6 words long. Although the sentences generated usually aren't grammatically correct as a full sentence, they actually make sense as a phrase. Grammar2 is the opposite in the sense that it has a fairly extensive list of rules, including the +nonterminants which allow for more diverse sentences, but they are all weighted equally. This means that the sentences generated vary in length and use more of the available vocabulary, but generally make very little sense as they are essentially generated by a unigram which fails to take into account the previous words when generating the current word. This is made more evident by the lack of a period at the end of the generated sentences. Combining the two actually does nothing relative to just using the first one since the rules defined in grammar1 are weighted much higher than the uniformly weighted rules defined in grammar2 so they all get used when the two get concatenated. Therefore, the resulting sentences look indistinguishable from those generated by just using the first ones rules.

Task 4
In designing my grammar, the first thing I did was look at some of the previous parse trees we studied in class and note what the most common derivations were and what exactly each of the POS tags meant. From there, I tried to model that with the probabilities I came up with. Next, after taking a look at the original lexicon, I decided to expand on it. I noticed the majority of words in there were labeled Misc, so I decided to

put some of them to use and label them as different categories such as ‘Pronoun’, ‘Adj’, and ‘Adverb’ in order to give my sentences more diversity. I then reflected these changes in my grammar rules to account for the discrepancies between categories and their use. After that, I focused on assigning different probabilities to different words in each of the categories based on how common I felt they were. After all that, I generated sentences and made minor modifications to the probabilities until I felt they were producing fairly realistic sentences.

Task 5

I felt as though 14 of my 20 generated sentences were grammatically correct, even if some of them didn’t make the most sense in context.