Alexander Soong
998857734
ECS189 HW3
March 5, 2018

Probabilistic Context Free Grammar

- **Task 1**
  - This PCFG implements a Bigram language model. One can derive this answer by examining *cfgparse.pl* that was provided to us. In this perl code, as the sentences are parsed, they are split by space characters and stored into either Binary or Unary structures. These two, Binary and Unary, structures were lists of lists that kept track of the previous or previous two words; hence a bigram language model. You can also tell by analyzing how the grammar is written. For each rule, the left side is constituted by at most 2 parts of speech.
- **Task 2**
  - ./cfgparse.pl grammar1 lexicon < examples.sen
  - ./cfgparse.pl grammar1 grammar2 lexicon < examples.sen
  - Above are the two commands that would create the parse trees for the sentences in *examples.sen* based on the grammars and lexicon that were provided to *cfgparse.pl*. The outputs of the two commands are significantly different. Of the 27 sentences provided in *examples.sen* all but 2 sentences resulted in *(failure)* when only grammar1 was provided. However, when both grammar1 and grammar2 were used to create the parses, it resulted in all 27 sentences being parse correctly.
  - This difference can be attributed to grammar1 being less exhaustive compared to grammar2. What I mean by exhaustive is there are certain rules that appear in the sentence structure from *examples.sen* that are not represented by the rules in grammar1, and they are more thoroughly represented in grammar2. So while the grammar provided in grammar1 are more grammatically correct, there are more flexible possibilities provided in grammar2. Also, grammar1 does not account for any of the "*Misc"* part of speech words in the lexicon, whereas grammar2 accounts for the "*Misc"* heavily.
- **Task 3**
  - ./cfggen.pl --text 10 grammar1 lexicon > sentence1.txt
  - ./cfggen.pl --text 10 grammar2 lexicon > sentence2.txt
  - ./cfggen.pl --text 10 grammar1 grammar2 lexicon > sentence3.txt
  - Above are the three commands that generated sentences based on the grammars provided and the lexicon. There is a clear differences in the sentences created by grammar1 and grammar2, and this difference can be attributed to the weights that are assigned to *ROOT.*
  - For grammar1, *ROOT -> S1* is given a weight of 99, and the only rule following that incorporates S1 is *S1 -> NP VP . 1* What this indicates is that sentences that are generated by grammar1 have a noun phrase followed by a verb phrase.
    - Some sentences generated by grammar1:

        - 1: every weight drinks each sovereign .
        - 2: each pound rides another husk .
        - 3: another home drinks each king .
        - 4: every story covers every weight .
        - 5: a corner drinks any sun .
    - If you analyze these sentences, you'll see that my claim was true. Nouns are followed by verbs.
        - weight (noun) drinks (verb)
        - Pound (noun) rides (verb)
        - Home (noun) drinks (verb)
        - Story (noun) covers (verb)
        - Corner (noun) drinks (verb)
    - For grammar2, the *ROOT -> S1* is given only a weight of 1. And the rules following show that S1 can lead to any of the provided parts of speech. Additionally, these other parts of speech can all lead to other parts of speech with no weight on particular rules. The effect of this is that the sentences generated using this grammar seem to not follow a solid structure and are grammatically incorrect.
    - For the last command, it is a mix of grammar1 and grammar2. Because there is structure enforced by the rules provided by grammar1, the sentences that are created usually follow the structure of at least having a noun phrase followed by a verb phrase. After that, because of the exhaustive list of rules provided by grammar2, the sentences that are created also exhibit some flexibility.
- **Task 4**
    - The first thing I did with the grammar was attempt to update the grammar so that there were no *Misc* included in it. I felt that this would give the grammar more structure instead of having every part of speech default to a *Misc.* The next thing I did was express additional parts of speech to the lexicon to account for all of the *Misc* that were present. I did this by using https://parts-of-speech.info/ to formulate a new list of parts of speech. Additional parts of speech tags included pronouns, adverbs, adjectives, punctuations, numbers, conjunctions among others. Lastly, based on my own knowledge of the English grammar, I placed weights on certain rules so that their effects would be reflected in the resulting parse. Such rules included, an adverb coming before a verb, verb coming before a noun/proper noun/pronoun, and conjunction joining two nouns. Eventually my cross entropy score for the parses was 74.76, whereas the cross entropy score for the combined grammar1 and grammar2 was 67.80.
- **Task 5**
    - My sentences and grammatically correct sentences are in the zip.
    - Of the 20 sentences generated, 13 seem to be grammatically correct.

I consulted with Henry Le on a general approach on how to formulate the grammar.