|  |  |
| --- | --- |
| ../../../../Desktop/barks.jpg  context free grammar  NLP Homework 3 | Komal Gujarathi  SUID - 211778351 |

Table of Contents

[1. Context Free Grammar 3](#_Toc510383188)

[2. Sample Sentences that can be parsed by this CFG 4](#_Toc510383189)

[3. Probabilistic Context Free Grammar 5](#_Toc510383190)

[4. Appendix 6](#_Toc510383193)

# 1. Context Free Grammar

**Develop a Context Free Grammar which can parse the following sentences. Use python environment to test your grammar to make sure it does the job and save python screen shots.**

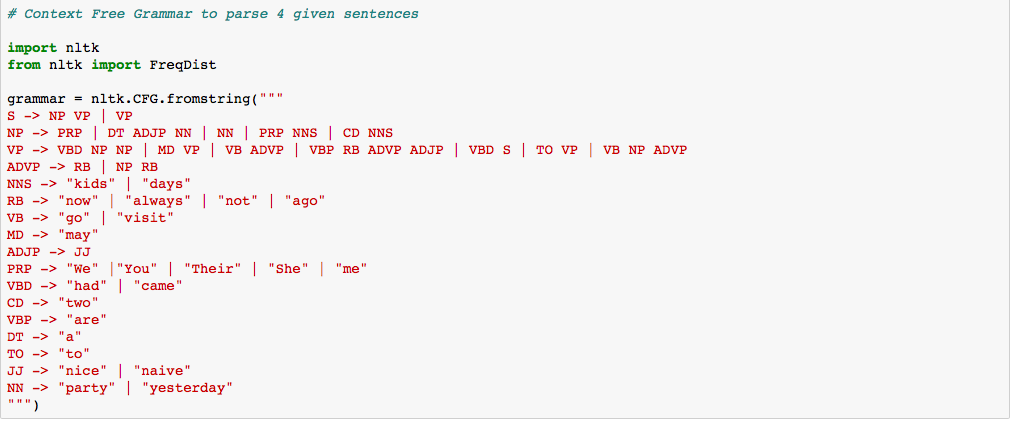
a. “We had a nice party yesterday”

b. “She came to visit me two days ago”

c. “You may go now”

d. “Their kids are not always naive”

Following is the grammar generated to parse the given sentences using recursive decent parser.



Please find the python processing screen shots in the appendix.

# 2. Sample Sentences that can be parsed by this CFG

**Besides these four sentences, list up to three different sentences that can be parsed by this grammar as well as the corresponding python file in the screen shot. Of these three sentences, can you generate one that does not make sense? Why can’t you have such a sentence based on your grammar?**

Three different sentences which can also be parsed using this grammar are –

1. She may visit now

2. You are not always nice

3. She came two kids yesterday

I have showed the python processing screen shots in the appendix. All the three sentences can be parsed by the recursive decent parser. The third sentence is parsed even if it doesn’t make sense. This is because Context Free Grammars can represent many parts of natural language adequately, but still has the problems like Agreement, Subcategorization, Movement.

The third sentence should actually be – “She came **with** two kids yesterday”, whereas our grammar could parse it even though the preposition ‘with’ was missing from the sentence. This also happens because, since when one rule derives to multiple different rules, then our parser can find the match to any rule, even though it might not make sense. In order to parse this meaningful sentence, we might need to add more rules to our grammar to parse the sentence.

# 3. Probabilistic Context Free Grammar

**Now, assuming that the above four sentences are your mini - mini training corpus, you will write a probabilistic context - free grammar. You will use Python environments to test your grammar and save the Python screenshot. (40%)**

# A Probabilistic Context-Free-Grammar is simply a CFG with probabilities assigned to the rules such that the sum of all the probabilities assigned to the rules such that the sum of all probabilities for all rules expanding the same non-terminal is equal to one.

# I have created a mini corpus of given 4 sentences as shown in the python processing screenshots. After creating this mini corpus, I have calculated the frequency counts in a corpus.

From the frequency distribution obtained from the words in this corpus, we can see that each word has the equal frequency of occurring, (that is 1 for each word). Thus, each word is equally likely to appear in the corpus.

So, while creating the probabilistic grammar, I have assigned the equal frequency to each terminal symbol.

For example, NN -> “party” | “yesterday”

In this, I have assigned the probability of 0.5 and 0.5 to each of these words, because each word is equally likely to appear in the text.

In order to decide the probability of the Non-terminal symbols we can analyze the text visually, We see that the rule S -> NP VP is used 90% of the times while parsing the sentences, and S -> VP is hardly used. Thus distribution –

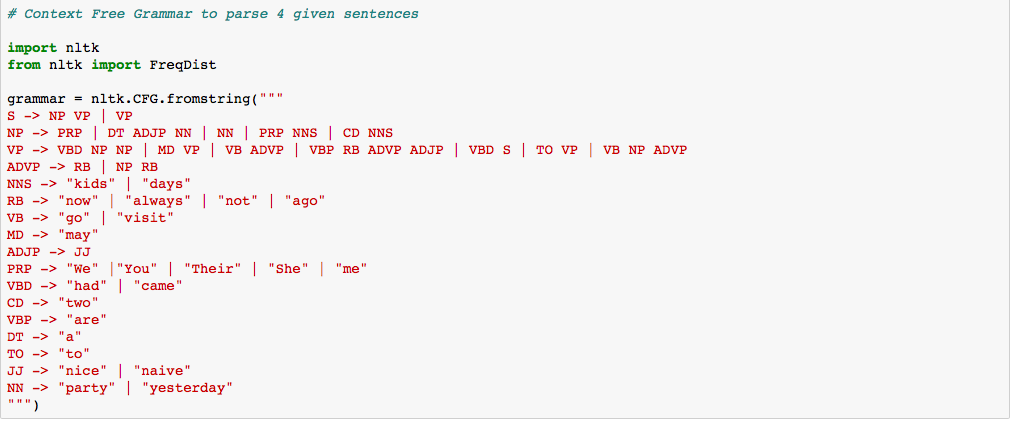
S -> NP VP[0.9] | VP[0.1]

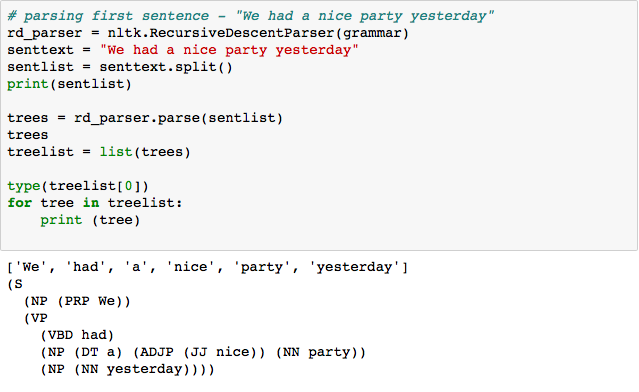
Only thing we need to make sure in this case is that the sum of the probabilities of all the non-terminal symbols must add up to 1.

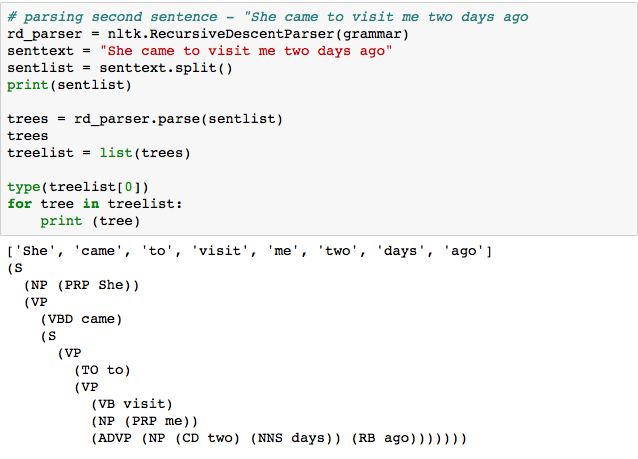
Please find the Grammar and the parsed sentences in the screenshots.

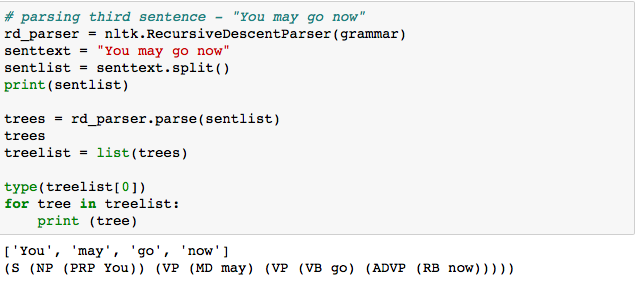
# 4. Appendix

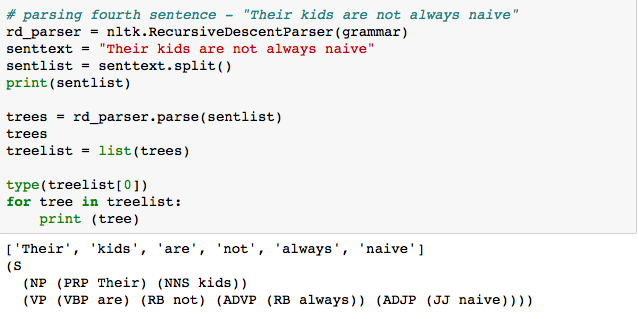
Code and Output

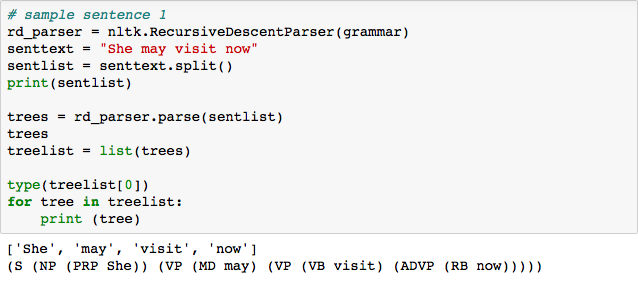


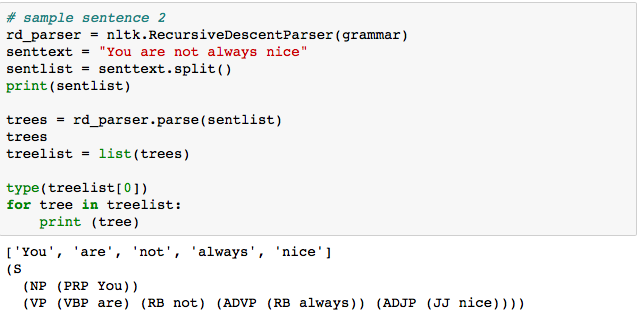


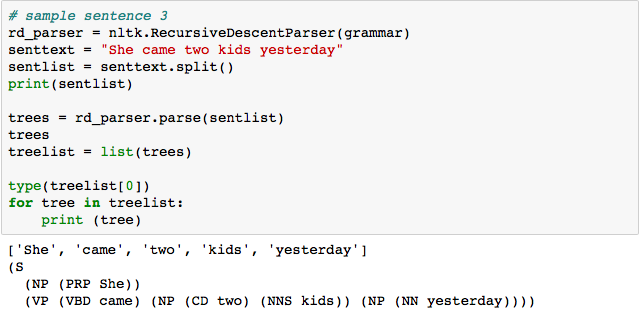


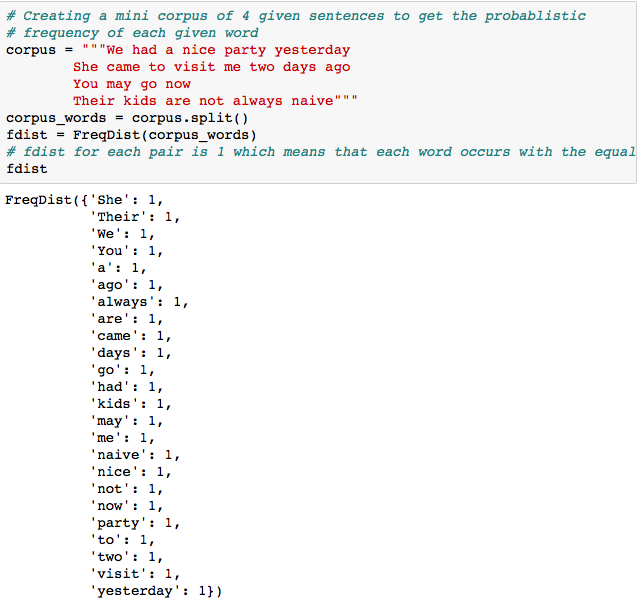


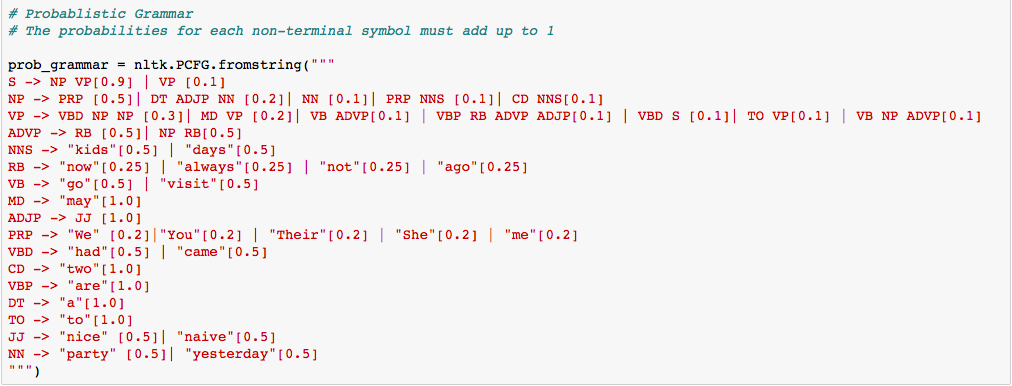












../../../../Desktop/Screen%20Shot%202018-04-01%20at%208.12.52%20PM.png

