HW3

Writing a CFG (Context-Free Grammar)

Bing-Je Wu

IST 664 – Natural Language Processing

Table of Contents

[Introduction 2](#_Toc40914301)

[Analysis and Models 2](#_Toc40914302)

[About the Data 2](#_Toc40914303)

[Data Processing 2](#_Toc40914304)

[Results 4](#_Toc40914305)

[Conclusion 12](#_Toc40914306)

[Reference 12](#_Toc40914307)

# Introduction

This assignment is to develop a CFG (context-Free Grammar) for a subset of English. There are two parts. The first part is to write grammar rules, with no option for programming, to parse a given set of sentences. The second part is to create two sampler sentences by using the same words from the given set of sentences. One made up sentence that uses the same grammar rules but cannot be parsed. The other sentence is able to be parsed but not an actual English sentence.

Analysis and Models  
A python program, development, provides the ability to parse sentences. Grammar rules and lexical rules need to be updated and rewritten in order to parse variety of sentences.

## About the Data

Two txt files were offered. One is regular English sentences. The other is challenging sentences with extra clauses, punctuation marks, wh- questions…etc.

Data Processing  
Some new rules were added such as, ‘adjective breakdown’, ‘noun breakdown’, ‘verb breakdown’, ‘proper noun breakdown’, ‘pronoun breakdown’ and ‘conjunction breakdown’. The breakdown rules help organize rules and show the rules in simplest form. Apart from the newly added breakdown rules, the Sentences Rule, Verb phrases Rule, Noun phrases rule and Prepositional phrase rules were updated with new alternative options and changing the tags for simplicity. A WH phrases Rule was added in order to parse advanced level of sentences from ‘challenge.sentences.txt’ file. The tags in the lexical rules were also changed to combine with breakdown rules.

***Table 1****. Grammar rules before and after*

|  |  |
| --- | --- |
| **Before update** | **After update** |
| # The start symbol is START. START -> S1  # Sentences S1 -> NP VP Eos  # Verb phrases VP -> VerbT NP | VerbT NP PP  # Noun phrases NP -> Det NP | Proper | Noun PP | Noun  # do not use Nominals in RD parser # Nom -> Noun | Noun Nom  # Prepositional phrase PP -> Prep NP | # The start symbol is START. START -> S1  # Sentences S1 -> NP VP Eos | VP Eos | WHP NP VP Eos  # WH phrases WHP -> WRB DO | WDT NP DO | WP DO  # Verb phrases VP -> V NP | V NP PP | V VP | V PP | V | V PP PP | V NP VP | V RB RB | V RB PP | V NOT ADJ  # Noun phrases NP -> Det NP | PropN | N PP | N | Det JJ NP | Pronoun | NUM N | NUM JJ NP | PropN C PropN | VBG PP  # Prepositional phrase PP -> Prep NP | TO NP | TO VP  ## Adjective breakdown # ADJ: adjective; JJR: comparative; JJS: superlative  JJ -> ADJ | JJR | JJS  ## Noun breakdown # NNS: noun pural; Noun: Singular and mass nouns N -> NNS | Noun  ## Verb breakdown # VerbT: transitive verbs; Vpast: past tense; VB: base form # VBN: past participle; VBG: present participles; VB3rd: third person singular; VBS: plural; DO: do/does V -> VerbT | Vpast | VB | VBN | VBG | VB3rd | VBS | MD | DO  ## Proper Noun breakdown # Proper: Proper nouns for people; NNP: proper nouns, not people  # ProperS: Plural proper nouns PropN -> Proper | NNP | ProperS  ## Pronoun breakdown # PProp: Personal pronouns; PPZ: Possessive personal pronouns Pronoun -> PProp | PPZ  ## Conjunction breakdown # CC: Coordinating conjunctions; SubC: Subordinating conjunctions C -> CC | SubC |

# Results

The sentence #1 to #16 cover the part 1 of the assignment. Two advanced sentences for part 1 were listed as sentence #17 and #18. Sentence #19 and #20 cover the part 2 of the assignment, one made up sentence that uses the same grammar rules but cannot be parsed and the other made up sentence that is able to be parsed but not an actual English sentence.

***Table 2****. Part 1 and Part 2*

|  |  |  |  |
| --- | --- | --- | --- |
| # | Sentence | Explain | Result |
| 1 | Arthur is the king . | There is only one tree for this sentence. This sentence. It used the ‘Sentence Rule’ and connect the ‘Proper Noun breakdown Rule’ from the ‘Noun phrase rule’  The ‘Proper’ tag was replaced by PropN to have the ‘Proper Noun breakdown Rule’ | (START  (S1  (NP (PropN (Proper Arthur)))  (VP (V (VerbT is)) (NP (Det the) (NP (N (Noun king)))))  (Eos .))) |
| 2 | Arthur rides the horse near the castle . | There are two trees for this sentence. The sentence can be parsed with nested ‘Noun Phrase’ in the ‘Verb Phrase’ or it can have only one ‘Noun Phrase’ in the ‘Verb Phrase’ | **1.** (START  (S1  (NP (PropN (Proper Arthur)))  (VP  (V (VerbT rides))  (NP  (Det the)  (NP  (N (Noun horse))  (PP (Prep near) (NP (Det the) (NP (N (Noun castle))))))))  (Eos .)))  **2.**(START  (S1  (NP (PropN (Proper Arthur)))  (VP  (V (VerbT rides))  (NP (Det the) (NP (N (Noun horse))))  (PP (Prep near) (NP (Det the) (NP (N (Noun castle))))))  (Eos .))) |
| 3 | Arthur rides the plodding horse near the castle . | Two trees were produced for this sentence. ‘NP -> Det JJ NP’ rule was added for this sentence. The first tree parse the ‘Verb phrase’ into ‘V NP’ However, the second tree parse the ‘Verb phrase’ into ‘ V NP PP’ at the third layer. | **1.**(START  (S1  (NP (PropN (Proper Arthur)))  (VP  (V (VerbT rides))  (NP  (Det the)  (JJ (ADJ plodding))  (NP  (N (Noun horse))  (PP (Prep near) (NP (Det the) (NP (N (Noun castle))))))))  (Eos .)))  **2.**(START  (S1  (NP (PropN (Proper Arthur)))  (VP  (V (VerbT rides))  (NP (Det the) (JJ (ADJ plodding)) (NP (N (Noun horse))))  (PP (Prep near) (NP (Det the) (NP (N (Noun castle))))))  (Eos .))) |
| 4 | the Holy\_Grail is a chalice . | There is only one tree for this sentence. The sentence was parsed into a ‘Noun Phrase’ and a ‘Verb Phrase’. A proper noun was used in the ‘Noun Phrase’. The ‘Verb Phrase’ in the second layer was broken-down with ‘V NP’. | (START  (S1  (NP (Det the) (NP (PropN (NNP Holy\_Grail))))  (VP (V (VerbT is)) (NP (Det a) (NP (N (Noun chalice)))))  (Eos .))) |
| 5 | the sensational Holy\_Grail is a sacred chalice . | ‘NNP -> Holy\_Grail’ lexcial rule was added for the sentence. ‘NP -> Det JJ NP’ rule was added for this sentence. The rule was used twice for this sentence when a ‘Noun Phrase’ needs to be broken down. | (START  (S1  (NP  (Det the)  (JJ (ADJ sensational))  (NP (PropN (NNP Holy\_Grail))))  (VP  (V (VerbT is))  (NP (Det a) (JJ (ADJ sacred)) (NP (N (Noun chalice)))))  (Eos .))) |
| 6 | every coconut was carried to the hottest mountains . | ‘PP -> TO NP’ rule was added for the sentence to parse the preposition, ‘to’. ‘NP -> Det JJ NP’ rule was also used to parse the sentence. | (START  (S1  (NP (Det every) (NP (N (Noun coconut))))  (VP  (V (Vpast was))  (VP  (V (Vpast carried))  (PP  (TO to)  (NP (Det the) (JJ (JJS hottest)) (NP (N (NNS mountains)))))))  (Eos .))) |
| 7 | sixty strangers are at the Round\_Table . | ‘NP -> NUMN’ rule was added to parse the numeric words or number words. ‘NP -> PropN’ was modified for the generalize the ‘Noun phrase’ rule. PropN -> NNP’ breakdown rule was added for the ‘Round\_Table. | (START  (S1  (NP (NUM sixty) (N (NNS strangers)))  (VP  (V (VBS are))  (PP (Prep at) (NP (Det the) (NP (PropN (NNP Round\_Table))))))  (Eos .))) |
| 8 | Sir\_Lancelot might have spoken . | ‘VP -> V VP’ rule was added and modified for this sentence. A breakdown rule, ‘V -> MD|VB|VBN’ was also added for parsing ‘might’, ‘have’ and ‘spoken’. There are two trees for this sentence. The reason is that the word, ‘have’ can be both ‘VB’ tag, base form, or ‘VBS’ tag, plural. | **1.** (START  (S1  (NP (PropN (Proper Sir\_Lancelot)))  (VP (V (MD might)) (VP (V (VB have)) (VP (V (VBN spoken)))))  (Eos .)))  **2.**(START  (S1  (NP (PropN (Proper Sir\_Lancelot)))  (VP (V (MD might)) (VP (V (VBS have)) (VP (V (VBN spoken)))))  (Eos .))) |
| 9 | Guinevere had been riding with Patsy for five weary nights . | ‘NP -> NUM JJ NP’ rule was added for parsing the phrase, ‘five weary nights’. ‘VP -> VBG PP’ was added for the phrase, ‘riding with Patsy’. There are 4 trees for this sentence for some reasons. The second and third trees were generated because the breakdown rule, ‘V -> VBG’. The word, ‘had’ also two tags, ‘Vpast’, past tense, and ‘VBN’, past participle. | **1.**(START  (S1  (NP (PropN (Proper Guinevere)))  (VP  (V (Vpast had))  (VP  (V (VBN been))  (NP  (VBG riding)  (PP (Prep with) (NP (PropN (Proper Patsy)))))  (PP  (Prep for)  (NP (NUM five) (JJ (ADJ weary)) (NP (N (NNS nights)))))))  (Eos .)))  **2.**(START  (S1  (NP (PropN (Proper Guinevere)))  (VP  (V (Vpast had))  (VP  (V (VBN been))  (VP  (V (VBG riding))  (PP (Prep with) (NP (PropN (Proper Patsy))))  (PP  (Prep for)  (NP (NUM five) (JJ (ADJ weary)) (NP (N (NNS nights))))))))  (Eos .)))  **3.**(START  (S1  (NP (PropN (Proper Guinevere)))  (VP  (V (VBN had))  (VP  (V (VBN been))  (NP  (VBG riding)  (PP (Prep with) (NP (PropN (Proper Patsy)))))  (PP  (Prep for)  (NP (NUM five) (JJ (ADJ weary)) (NP (N (NNS nights)))))))  (Eos .)))  **4.**(START  (S1  (NP (PropN (Proper Guinevere)))  (VP  (V (VBN had))  (VP  (V (VBN been))  (VP  (V (VBG riding))  (PP (Prep with) (NP (PropN (Proper Patsy))))  (PP  (Prep for)  (NP (NUM five) (JJ (ADJ weary)) (NP (N (NNS nights))))))))  (Eos .))) |
| 10 | Sir\_Bedevere might have been suggesting this quest . | ‘V -> MD|VB|VBS|VBN’ breakdown rule was added for parsing ‘might’, ‘have’, ‘been’ and ‘suggesting’. There two trees for this sentence. The word, ‘have’, have both ‘VB’ tag, base form, and ‘’VBS’ tag, plural. The trees have three nested ‘Verb Phrase’ with in the ‘Verb Phrase’ at the second layer. | **1.**(START  (S1  (NP (PropN (Proper Sir\_Bedevere)))  (VP  (V (MD might))  (VP  (V (VB have))  (VP  (V (VBN been))  (VP  (V (VBG suggesting))  (NP (Det this) (NP (N (Noun quest))))))))  (Eos .)))  **2.**(START  (S1  (NP (PropN (Proper Sir\_Bedevere)))  (VP  (V (MD might))  (VP  (V (VBS have))  (VP  (V (VBN been))  (VP  (V (VBG suggesting))  (NP (Det this) (NP (N (Noun quest))))))))  (Eos .))) |
| 11 | the Britons migrate south frequently . | ‘VP -> V RB RB’ rule was added for the phrase, ‘migrate south frequently’. The word ‘migrate’ has two tags, ‘VB’ tag and ‘VBS’ tag. Therefore, there are two trees for this sentence. | **1.**(START  (S1  (NP (Det the) (NP (PropN (ProperS Britons))))  (VP (V (VB migrate)) (RB south) (RB frequently))  (Eos .)))  **2.**(START  (S1  (NP (Det the) (NP (PropN (ProperS Britons))))  (VP (V (VBS migrate)) (RB south) (RB frequently))  (Eos .))) |
| 12 | Arthur and Guinevere ride frequently near the castle . | ‘NP -> PropN C PropN’ was added for the phrase, ‘Arthur and Guinevere’. ‘VP -> V RB PP’ rule was added for parsing ‘ride frequently near the castle’. ’There are two trees for this sentence because the word, ‘ride’, has two tags, ‘VB’ tag, base form, and ‘’VBS’ tag, plural. | **1.**(START  (S1  (NP  (PropN (Proper Arthur))  (C (CC and))  (PropN (Proper Guinevere)))  (VP  (V (VB ride))  (RB frequently)  (PP (Prep near) (NP (Det the) (NP (N (Noun castle))))))  (Eos .)))  **2.**(START  (S1  (NP  (PropN (Proper Arthur))  (C (CC and))  (PropN (Proper Guinevere)))  (VP  (V (VBS ride))  (RB frequently)  (PP (Prep near) (NP (Det the) (NP (N (Noun castle))))))  (Eos .))) |
| 13 | he suggests to grow fruit at home . | ‘PP -> TO VP’ rule was added for the phrase, ‘to grow fruit at home’. There are 6 trees for this sentence. The word, ‘grow’ has two tags, ‘VB’ tag, base form, and ‘’VBS’ tag, plural. The ‘Verb Phrase’ can be broken down as ‘ V PP’, ‘V NP’ and ‘V PP PP’. There are two prepositions, ‘to’ and ‘at’ in the sentence. ‘VP -> V PP PP’ and ‘VP -> V NP’ rules were added for dealing with two prepositions. | **1.**(START  (S1  (NP (Pronoun (PProp he)))  (VP  (V (VB3rd suggests))  (PP  (TO to)  (VP  (V (VB grow))  (NP (N (Noun fruit)) (PP (Prep at) (NP (N (Noun home))))))))  (Eos .)))  **2.**(START  (S1  (NP (Pronoun (PProp he)))  (VP  (V (VB3rd suggests))  (PP  (TO to)  (VP  (V (VBS grow))  (NP (N (Noun fruit)) (PP (Prep at) (NP (N (Noun home))))))))  (Eos .)))  **3.**(START  (S1  (NP (Pronoun (PProp he)))  (VP  (V (VB3rd suggests))  (PP  (TO to)  (VP  (V (VB grow))  (NP (N (Noun fruit)))  (PP (Prep at) (NP (N (Noun home)))))))  (Eos .)))  **4.**(START  (S1  (NP (Pronoun (PProp he)))  (VP  (V (VB3rd suggests))  (PP  (TO to)  (VP  (V (VBS grow))  (NP (N (Noun fruit)))  (PP (Prep at) (NP (N (Noun home)))))))  (Eos .)))  **5.**(START  (S1  (NP (Pronoun (PProp he)))  (VP  (V (VB3rd suggests))  (PP (TO to) (VP (V (VB grow)) (NP (N (Noun fruit)))))  (PP (Prep at) (NP (N (Noun home)))))  (Eos .)))  **6.**(START  (S1  (NP (Pronoun (PProp he)))  (VP  (V (VB3rd suggests))  (PP (TO to) (VP (V (VBS grow)) (NP (N (Noun fruit)))))  (PP (Prep at) (NP (N (Noun home)))))  (Eos .))) |
| 14 | riding to Camelot is not hard . | ‘NP -> VBG PP’ rule was added for the phrase, ‘ridding to Camelot’. ‘VP -> V NOT ADJ’ was added for parsing ‘is not hard’. | (START  (S1  (NP (VBG riding) (PP (TO to) (NP (PropN (NNP Camelot)))))  (VP (V (VerbT is)) (NOT not) (ADJ hard))  (Eos .))) |
| 15 | do coconuts speak ? | ‘S1 -> VP Eos’ rule was added for the question sentence. ‘VP -> V NP VP’ was added for parsing ‘do coconuts speak’. There are two trees for this sentence because the word, ‘speak’ has two tags, ‘VB’ tag, base form, and ‘’VBS’ tag, plural. | **1.** (START  (S1  (VP (V (DO do)) (NP (N (NNS coconuts))) (VP (V (VB speak))))  (Eos ?)))  **2.**(START  (S1  (VP (V (DO do)) (NP (N (NNS coconuts))) (VP (V (VBS speak))))  (Eos ?))) |
| 16 | why does England have a king ? | ‘S1 -> WHP NP VP Eos’ rule was added for the question sentence. ‘WHP -> WRB DO’ rule was created for parsing ‘why does’. There are two trees for this sentence because the word, ‘have’ has both ‘VB’ tag, base form, and ‘’VBS’ tag, plural. | **1.**(START  (S1  (WHP (WRB why) (DO does))  (NP (PropN (NNP England)))  (VP (V (VB have)) (NP (Det a) (NP (N (Noun king)))))  (Eos ?)))  **2.**(START  (S1  (WHP (WRB why) (DO does))  (NP (PropN (NNP England)))  (VP (V (VBS have)) (NP (Det a) (NP (N (Noun king)))))  (Eos ?))) |
| 17 | what horse does Arthur ride ? | ‘S1 -> WHP NP VP Eos’ rule was added for the question sentence. ‘WHP -> WDT NP DO’ was created for the phrase, ‘what horse does’. There are two trees for this sentence because the word, ‘ride’ has both ‘VB’ tag, base form, and ‘’VBS’ tag, plural. | **1.**(START  (S1  (WHP (WDT what) (NP (N (Noun horse))) (DO does))  (NP (PropN (Proper Arthur)))  (VP (V (VB ride)))  (Eos ?)))  **2.**(START  (S1  (WHP (WDT what) (NP (N (Noun horse))) (DO does))  (NP (PropN (Proper Arthur)))  (VP (V (VBS ride)))  (Eos ?))) |
| 18 | who does Arthur suggest she carry ? | ‘S1 -> WHP NP VP Eos’ rule was added for the question sentence. ‘WHP -> WP DO’ rule was created for parsing ‘who does’. ‘VP -> V NP VP’ was added for the phrase, ‘suggest she carry’.  There are four trees for this sentence because the words, ‘suggest’ and ‘carry’ have both ‘VB’ tag, base form, and ‘’VBS’ tag, plural. | **1.**(START  (S1  (WHP (WP who) (DO does))  (NP (PropN (Proper Arthur)))  (VP  (V (VB suggest))  (NP (Pronoun (PProp she)))  (VP (V (VB carry))))  (Eos ?)))  **2.**(START  (S1  (WHP (WP who) (DO does))  (NP (PropN (Proper Arthur)))  (VP  (V (VB suggest))  (NP (Pronoun (PProp she)))  (VP (V (VBS carry))))  (Eos ?)))  **3.**(START  (S1  (WHP (WP who) (DO does))  (NP (PropN (Proper Arthur)))  (VP  (V (VBS suggest))  (NP (Pronoun (PProp she)))  (VP (V (VB carry))))  (Eos ?)))  **4.**(START  (S1  (WHP (WP who) (DO does))  (NP (PropN (Proper Arthur)))  (VP  (V (VBS suggest))  (NP (Pronoun (PProp she)))  (VP (V (VBS carry))))  (Eos ?))) |
| 19 | do not drink again ! | Although ‘S1 -> VP Eos’ rule allow a sentence without having a ‘Noun Phrase’ in the beginning to make a legal sentence. The phrase, ‘do not drink again’ cannot be parsed by current grammar. A new rule ‘VP -> V NOT VP RB’ should be added in order to parse this sentence. | **There is no tree for this sentence.** |
| 20 | Arthur does drunk with Patsy . | Although the sentence was parsed successfully, it is not a correct English sentence. The word, ‘drunk’, should be ‘drink’ since it is following the auxiliary verb, ‘do’. Any verb follows the auxiliary verb should be in simple present. | (START  (S1  (NP (PropN (Proper Arthur)))  (VP  (V (DO does))  (VP  (V (VBN drunk))  (PP (Prep with) (NP (PropN (Proper Patsy))))))  (Eos .))) |

# Conclusion

In the part one of assignment, most of the sentences can be parsed with specific grammar rules and lexical rules. Some may have several results generated by the parser because of the general rules or common lexical tags. From the programming perspective, the CFG should be general for parsing different sentences. However, the idea could be dangerous, and it can be having the issue so called ‘overgeneralized’. We can see from the part two of assignment. The auxiliary verb should be followed by a verb in the simple present form. In the example of the sentence #20, it was parsed incorrectly. Thus, when we create the CFGs, we need to consider not only the general rule for parsing sentences but also the restricted rule of English grammar.

# Reference

1 Part-of-speech (POS) annotation. [Web Link](https://www.ling.upenn.edu/~beatrice/annotation-for-audio-aligned-corpora/pos.html) ;