Natural Language Processing ECS763P

Course Work 2: Formal Grammars and Parsing

Based on chapters 11,12,13 of Jurefsky and Martin

Posted on February 18th 2018, Due on 11th of March 2018

Please submit via QMPlus.

1. Use the rules provided in Chapter 11 of the Jurafsky and Martin book as guidelines to draw CFG parse trees for the following phrases. In each case, write down the rules you used. For the rules that are mentioned not in the book or on the slides, you can use the parses of the ATIS corpus to get ideas.
   (a) Baltimore
   (b) to Baltimore
   (c) near Baltimore
   (d) the flight
   (e) American's flights
   (f) morning flight
   (g) one flight
   (h) any flight
   (i) any morning flight from Baltimore
   (j) United's flight to Baltimore
   (k) any flight leaving from Baltomore
   (l) any flight that serves lunch
   (m) United's pilot's protests
2. Do the same as in Question 1, following the same procedure, but now for the following sentences:
   (a) Do American Airlines have a flight between Baltimore and Denver?
   (b) I would like to fly on American Airlines.
   (c) Please repeat that.
   (d) I need to fly between Philadelphia and Atlanta.
   (e) What is the fare from Atlanta to Denver?
   (f) We flew to Baltimore and Denver.
3. The provided script (`grammar_script.py`) contains three parsed sentences from the Penn Treebank. Using this script, draw the CFG parse trees for these sentences and extract the set of CFG rules that are needed to parse the original sentences.
4. Consider the following sentence:

   "List me the seats on the flight to Denver"

   (a) Give as many meanings for this sentence as you can (aim for more than 3).

(b) Replace the grammar in the `grammar_script.py` file with the rules of the miniature grammar of English given in Figure 12.1, plus the rules listed below, as well as any extra lexicon rules necessary to parse the above sentence. Find all possible parses of the above sentence.

Note: in the rules below, IVP stands for imperative verb phrase, and IVerb stands for imperative verb.

$$
\begin{aligned}
S &\rightarrow IVP \\
IVP &\rightarrow IVerb\ NP\ NP \\
IVP &\rightarrow IVerb\ NP\ NP\ PP
\end{aligned}
$$

Grammar	Lexicon
S → NP VP	*Det → that | this | the | a*
S → Aux NP VP	*Noun → book | flight | meal | money*
S → VP	*Verb → book | include | prefer*
NP → Pronoun	*Pronoun → I | she | me*
NP → Proper-Noun	*Proper-Noun → Houston | NWA*
NP → Det Nominal	*Aux → does*
Nominal → Noun	*Preposition → from | to | on | near | through*
Nominal → Nominal Noun	
Nominal → Nominal PP	
VP → Verb	
VP → Verb NP	
VP → Verb NP PP	
VP → Verb PP	
VP → VP PP	
PP → Preposition NP	

Figure 12.1 The $\mathscr{L}_1$ miniature English grammar and lexicon.

(c) Repeat the above procedure, but this time with the following rule added to your previous stack of rules:

$$NP \rightarrow NP\ PP$$

How many new parses do you get this time?

(d) Did you discover any new meanings from the parses of parts (b) and (c) above? Describe and discuss these new meanings. List any parses that are nonsensical.

(e) Use the following dependency labels table and without using any NLTK routines, so by hand, transform the CFG parse trees provided by NLTK for part (b) above (so before adding the rule in part(c)) to dependency trees.

Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

Figure 14.2 Selected dependency relations from the Universal Dependency set. (de Marneffe et al., 2014)

5. Using the syntactic categories and the lexicon expemlified in Ch. 11, draw a CCG parse tree for 2.(f). Clearly label your tree with the CCG rules that you are using.

6. Load the parsed version of ATIS corpus (`parseTrees.txt`) into NLTK using the provided script (`grammar_prob.py`) and extract the CFG grammar used to parse it.

   Learn the probabilities of your grammar from the parsed corpus and turn your grammar into a PCFG.

   Now, given your PCFG, compute what is the most likely parse of the ambiguous sentence

   "Show me the meals on the flight from Phoenix"

   Provide the following two items with your answer to this question:

   (a) your PCFG

   (b) the probabilities of each of the parse trees of the sentence.