Assignment 5

CS 421: Natural Language Processing

Due: October 30, 2020 (12 p.m. CST)

1. Introduction

In this assignment, you will learn about syntactic parsing, dependency relations, and part-of-speech tagging.

Syntactic parsing is the process of automatically converting natural language input into a structured hierarchical representation based on its grammatical constituents. There are many ways to produce syntactic parses, ranging from naive strategies to sophisticated dynamic programming or search methods. One of the most popular approaches for constituency parsing is the *probabilistic CKY algorithm*, a dynamic programming approach that seeks to resolve ambiguities by predicting the most likely parse tree for a given input. You can read more about constituency parsing in Chapters 13 and 14 of the course textbook.

Dependency relations are useful in information extraction, semantic parsing, and other NLP applications, and dependency parsing is particularly suitable for languages with free word order. You may read more about dependency parsing in Chapter 15 of the course textbook, and about the comparison between constituency and dependency relations on this website.[1]

Ambiguity occurs in natural language when multiple valid representations (such as constituency trees or dependency parses) can be built for the same input—in other words, when the same input can be interpreted in multiple ways. There are many types of ambiguities in natural languages, including but not limited to lexical, syntactic, and semantic ambiguities. Chapters 13-15 of the course textbook describe numerous types of ambiguity.

Please follow the instructions in Section 2 to solve all the questions listed in Section 3.

2. Instructions

Each question is labeled as `Code` or `Written` and the guidelines for each type are provided below.

[1] `http://www.ilc.cnr.it/EAGLES96/segsasg1/node44.html`

Code

Any `Code` questions need to be completed using Python (version 3.6+). If you want to use any external packages, you are required to get approval from the course staff on Piazza prior to submission. Templates are provided for each `Code` question (.py files) as supplementary material. Do not rename/delete any functions or global variables provided in these templates and write your solution in the specified sections. Use the `main` function (provided in templates) to test your code when running it from a terminal. Avoid writing that test code in the global scope, however, you should write additional functions/classes as needed in the global scope. These templates may also contain important information and/or examples in comments so please read them carefully. This part of the assignment will be graded automatically using Gradescope.

To submit your solution for `Code` questions, you need to compress the following files (after completion) in a single `zip` file. These files should be in the root of your `zip` archive for autograding to work correctly.

- ☐ `q2.csv`

Submit this `zip` file on Gradescope under `Assignment 4 - Code`. All specified files need to be submitted to receive full credit.

Written

You are required to submit all `Written` questions in a single PDF file. You may create this PDF using Microsoft Word, scans of your handwritten solution, LaTeX or any other method you prefer.

To submit your solution for `Written` questions, you need to provide answers to the following questions in a single PDF.

- ☐ Q1
- ☐ Q3

Before submission, ensure that all pages of your solution are present and in order. Submit this PDF on Gradescope under `Assignment 4 - Written`. Please match all questions to their respective solutions (pages) on Gradescope. Questions not associated with any pages will be considered blank or missing and all questions need to be completed to receive full credit.

3. Questions

Q1 (60): Written

Use the probabilistic CKY algorithm to parse the following sentence (ignore punctuation): *Dracula mixed the Halloween candy with chopsticks.* You must show a complete table like the one in Figure 14.4 of the course textbook or

the "Probabilistic CKY Algorithm" video from class. Use the following grammar rules:

S → NP VP (60%)
S → N VP (20%)
S → NP V (20%)

VP → VP P (5%)
VP → VP PP (30%)
VP → VP NP (20%)
VP → VP N (10%)
VP → V P (5%)
VP → V N (10%)
VP → V PP (10%)
VP → V NP (10%)

NP → DT NP (20%)
NP → DT N (15%)
NP → NP N (5%)
NP → NP NP (10%)
NP → NP PP (20%)
NP → N N (5%)
NP → N NP (5%)
NP → N PP (20%)

PP → P N (40%)
PP → P NP (60%)

DT → the (100%)
P → with (100%)
V → mixed (75%)
V → candy (25%)
N → Dracula (25%)
N → Halloween (25%)
N → candy (25%)
N → chopsticks (25%)

Supplementary material: NA

Q2 (20): Code

Assign dependency relations to the following sentence, using the universal dependencies tagset:[2]

Dracula mixed the Halloween candy with chopsticks

[2] https://universaldependencies.org/u/dep/index.html

Submit your solution as a `.csv` file named `q2.csv` such that the first column (column 0) contains a valid dependency relation (`nsubj`, `obj`, etc.) and the second and third columns contain valid words from the sentence (`Halloween`, `candy`, etc.) among which the relationship exists. Since the dependency relations are directional, the direction is assumed to be from the word in the second column (head) to the word in the third column (dependent). There is no need to specify `root`. For reference, dependency relations (`denverflight.csv`) for the example sentence 15.1 from the course textbook are provided.
Supplementary material: `denverflight.csv`

Q3 (20): Written

Search through your favorite news source (social media is acceptable, but please anonymize identities that are not public) for three examples of ambiguous sentences or headlines. Describe the sources of the ambiguities (lexical, syntactic, etc.) for each example, and brainstorm at least one potential way you might be able to handle each ambiguity computationally.
Supplementary material: `NA`

4. Rubric

This assignment will be graded according to the rubric below. Partial points may be awarded for rubric items at the discretion of the course staff.

Q1 (60 points possible)	
Probabilistic CKY table assigns correct constituents and probabilities in the diagonal from (0,0) to (6,6)	+14
Probabilistic CKY table assigns correct constituents and probabilities in the diagonal from (0,1) to (5,6)	+12
Probabilistic CKY table assigns correct constituents and probabilities in the diagonal from (0,2) to (4,6)	+10
Probabilistic CKY table assigns correct constituents and probabilities in the diagonal from (0,3) to (3,6)	+8
Probabilistic CKY table assigns correct constituents and probabilities in the diagonal from (0,4) to (2,6)	+6
Probabilistic CKY table assigns correct constituents and probabilities in the diagonal from (0,5) to (1,6)	+6
Probabilistic CKY table assigns correct constituents and probabilities in cell (0,6)	+4
Q2 (20 points possible)	
(Autograded)	
Q3 (20 points possible)	
Ambiguity #1 is provided	+1
Source of ambiguity #1 is described in 1-2 sentences	+2
Potential way to address ambiguity #1 is explained in 1-2 sentences	+4
Ambiguity #2 is provided	+1
Source of ambiguity #2 is described in 1-2 sentences	+2
Potential way to address ambiguity #2 is explained in 1-2 sentences	+4
Ambiguity #3 is provided	+1
Source of ambiguity #3 is described in 1-2 sentences	+2
Potential way to address ambiguity #3 is explained in 1-2 sentences	+3