START OF QUIZ Student ID: 91905067,Nongbualang,Bon

Question 1

Topic: Lecture 2
Source: Lecture 2

Why do we not use accuracy to evaluate chunkers? Can you think of any other tasks where this might be as big (or bigger) of a problem? (1)

Question 2

Topic: Lecture 3
Source: Lecture 3

Explain why the following rule is not valid in a CFG: NP VB -> DT NN VB (1)

Question 3

Topic: Lecture 1
Source: Lecture 1

Imagine we were trying to create a treebank for an unknown language. We start by creating a list of words with their parts of speech. Do you think it would make sense to collect open or closed classes first? Explain. (1)

Question 4

Topic: Lecture 3
Source: Lecture 3

What is meant by ”context-free” in a context-free grammar? Why is this sufficient to represent most language, which we know is very contextually dependent? (1)

Question 5

Topic: Lecture 2
Source: Lecture 2

Briefly describe chunking, why it's easier than parsing, and why it is an important task in NLP. (1)

Question 6

Topic: Lecture 4
Source: Lecture 4

Given two parse trees, calculate the PARSEVAL score. Also briefly describe whether any errors are "syntacto-semantic" errors (ie, an error that requires real-world knowledge to arrive at the correct parse). 1: (S (NP (DT The) (JJ quick) (JJ brown) (NN fox)) (VP (VBZ jumps) (PP (IN over) (NP (DT the) (JJ lazy) (NN dog))))) (2): (S (NP (NP (DT The) (NP (JJ quick) (NP (JJ brown) (NN fox))))) (VP (VBZ jumps) (PP (IN over) (NP (DT the) (JJ lazy) (NN dog)))))

Question 7

Topic: Lecture 4
Source: Lecture 4

Basque is an ”ergative-absolutive” language - instead of defining NPs with respect to labels such as ”subject” and ”direct object”, NPs are defined with respect to ”subject of a transitive verb” (ergative) or ”subject of an intransitive verb OR object of a transitive verb” (absolutive). Explain what features would need to be defined in such a grammar, and how they would interact (you can assume a similar SVO order as English). (2)

Question 8

Topic: Lecture 1
Source: Lecture 1

We use trees to represent the structure of a parse, but that doesn't necessarily mean we have to use a Python Tree to represent them. Can you think of an alternative way of representing a syntax tree, preserving the hierarchy and traversal features inherent in a tree (no, you can't just create a "Shrub" class). Write some pseudocode that shows how this structure works. (2)

Question 9

Topic: Long
Source: Lecture 1

Imagine you're a text-to-speech (TTS) engineer. You've been asked by your boss to make your system sound more authentic by incorporating intonation into your model. Intonation is a pitch and stress pattern that differs between different pragmatic conditions. For example, English yes-no questions have a rising pitch on the end of the clause, imperative statements (ie, commands) have a falling pitch, and declarative sentences, while also falling, are not typically as sharp a fall as imperative sentences. How might you use this information, along with a parser, to modify your TTS system? Are there any complications or ambiguities that you can think of? (3)

END OF QUIZ