

START OF QUIZ

Student ID:

55343529,Nadal,Jacob

Question 1

Topic: Lecture 7

Source: Lecture 7

Briefly describe how the stack changes for a SHIFT operation. (1)

Question 2

Topic: Lecture 5

Source: Lecture 5

Why do we only care about the "upper triangle"? (1)

Question 3

Topic: Lecture 6

Source: Lecture 6

In class, we mentioned that the Earley and CYK parsers are both cubic complexity, but that in practice, the Earley Parser is typically faster. Why do you think that is? (1)

Question 4

Topic: Lecture 5

Source: Lecture 5

What is fenceposting? Give two reasons we need it in the CYK algorithm. (1)

Question 5

Topic: Lecture 7

Source: Lecture 7

What is a projective sentence? Why does this matter for the shift-reduce algorithm? (1)

Question 6

Topic: Lecture 6

Source: Lecture 6

What difficulties do you envision when using the Earley parser with a language with large amounts of agreement? (2)

Question 7

Topic: Lecture 8

Source: Lecture 8

Imagine that we have a dependency parser that has a very good UAS (90+), but a very bad LAS (50-). Do you think that we could use the output of this parser as input to a neural translation model as is, or do you think that we should first re-train the labeling part of the algorithm to increase LAS? Doing both is probably the best solution, but I'm asking if you think that we could use the output of the existing model, even as we try to improve the quality of the labels. Explain. (2)

Question 8

Topic: Lecture 8

Source: Lecture 8

In class, we mentioned that graph-based parsing can handle non-projective parses, but it has cubic time complexity. How would you go about improving the complexity to (mostly) linear time, while still being able to handle non-projective parses? Describe why this solution works. Hint: we talked about a simple solution in class. (2)

Question 9

Topic: Long

Source: Lecture 8

Let's say you have a friend who is developing a constructed language (conlang) for the epic fantasy novel he is writing, but the only language she knows is English, and she is just doing a word-for-word translation of English into this constructed language. She has no real knowledge of syntax, and has only ever thought that "some words come after other words" (she's thinking like a language model). How would you use treebanks and dependency parsers to demonstrate to her that there is a whole "hidden" structure that language must follow, and how would you help her develop a realistic grammar for her conlang? Do you think it would make more sense to build the grammar for a "modified English", and then do word-for-word translation, or to translate the words from English, and then build the grammar in the conlang? (3)

END OF QUIZ