

START OF QUIZ

Student ID:

92642370,Chen,Tiffany

Question 1

Topic: Lecture 3

Source: Lecture 3

Explain how phrasal attachment errors produce ambiguity. (1)

Question 2

Topic: Lecture 3

Source: Lecture 3

Explain why the following rule is not valid in a CFG: $\text{dog VB} \rightarrow \text{dog barks}$ (1)

Question 3

Topic: Lecture 4

Source: Lecture 4

Imagine that you are a comedian writing jokes. How might you use an automatic parser to help you find material? Briefly explain. (1)

Question 4

Topic: Lecture 4

Source: Lecture 4

Name 2 advantages of feature grammars over CFGs, and briefly explain why they are useful.

(1)

Question 5

Topic: Lecture 1

Source: Lecture 1

English is an SVO (Subject-Verb-Object) language, but only most of the time. Can you think of an example where this order is violated? Why do you think this doesn't confuse speakers of English? (1)

Question 6

Topic: Lecture 2

Source: Lecture 2

Imagine you're working on analysing customer feedback, and your boss wants you to identify the most common complaints. How might you use your parsing knowledge to automate and distill the most common complaints? You can assume that complaints have already been labeled with the product they are complaining about. You can also assume that just sorting the frequency of tokens is going to be insufficient. (2)

Question 7

Topic: Lecture 2

Source: Lecture 2

Do you think that we could do dependency parsing and a constituency-based task (such as chunking) at the same time? What features of the tasks might support each other (additive qualities), and which might make such a task more difficult (adversarial qualities)? (2)

Question 8

Topic: Lecture 1

Source: Lecture 1

Imagine someone develops a new parser that has 100

Question 9

Topic: Long

Source: Lecture 3

In class, we briefly mentioned OSASCOMP (the order of adjectives in English - Opinion, Size, Age, Shape, Colour, Origin, Material, Purpose). For example, we can have the "big red Italian car", but not the "red Italian big car". Please compose a CFG that can handle this ordering (you can assume that our grammar already knows what adjectives and noun phrases are). (3)

END OF QUIZ