CSE556: NLP - Mid-Sem

Name: _____ **Roll:** _____

Marks: 40
Duration: 1 hour **Date:** 26-Feb-2024

1. Describe the purpose of continuation probability. Also, write the equation for the same. **[2+2]**

Continuation probability is used when a token has a high unigram probability but it mostly occurs with another specific token. Thus, the continuation probability determines the likelihood of a token as the continuation of the previous token. For example, the token *"francisco"* will have a higher continuation probability if the previous token is "San" and relatively lower for other tokens. (2 marks)

$$P_{Continuation}(w_i) = \frac{|\{w_{i-1} : count(w_{i-1}w_i) > 0\}|}{|\{(w_{j-1}w_j) : count(w_{j-1}w_j) > 0\}|}$$

where $(w_{j-1}w_j)$ is all possible bigrams. (2 marks)

2. What do you mean by local vs distributed word representations? **[2]**

In local word representation, the meaning/semantic of the word is restricted to a single dimension of the vector. (1 mark)
In distributed word representation, the meaning/semantic of the word is spread across multiple dimensions of the vector. (1 mark)

3. Design a single and appropriate grammar for the following two sentences. Show the derivation or the parse tree utilizing the same grammar. **[4+2+2]**
   a. John ate rice with curd.
   b. John ate rice with a spoon.

Grammar G: <V, T, P, S>
V = {S, NN, VBD, DT, IN, NP, VP, PP}
T = {John, ate, rice, with, a, spoon, curd}
S = {S}
P:

S → NP VP NP → NN NP → DT NN NP → NP PP PP → IN NP VP → VBD NP VP → VBD NP PP	IN → with VBD → ate DT → a NN → John NN → rice NN → spoon NN → curd

(4 marks)

Note:

- An equivalent grammar is also fine as long as it generates at least one parse tree of each case, e.g., grammar symbols are different (verb for VBD, noun for nn), production rules are different.
- Grammar can be ambiguous but it should generate at least one parse tree of each case.

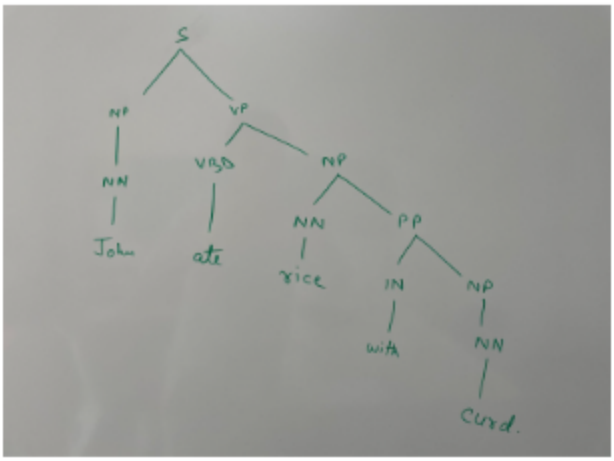
a. John ate rice with curd (**Either of the following for each case**) (2 marks)

Top-down derivation: S ⇒ **NP** VP ⇒ **NN** VP ⇒ John **VP** ⇒ John **VBD** NP ⇒ John ate **NP** ⇒ John ate **NN** PP ⇒ John ate rice **PP** ⇒ John ate rice **IN** NP ⇒ John ate rice with **NP** ⇒ John ate rice with **NN** ⇒ John ate rice with curd

OR

Bottom-up derivation: John ate rice with curd ⇒ **NN** ate rice with curd ⇒ NP **ate** rice with curd ⇒ NP VBD **rice** with curd ⇒ NP VBD NN **with** curd ⇒ NP VBD NN IN **curd** ⇒ NP VBD NP IN **NN** ⇒ NP VBD NP **IN NP** ⇒ NP VBD **NN PP** ⇒ NP **VBD NP** ⇒ **NP VP** ⇒ S

OR



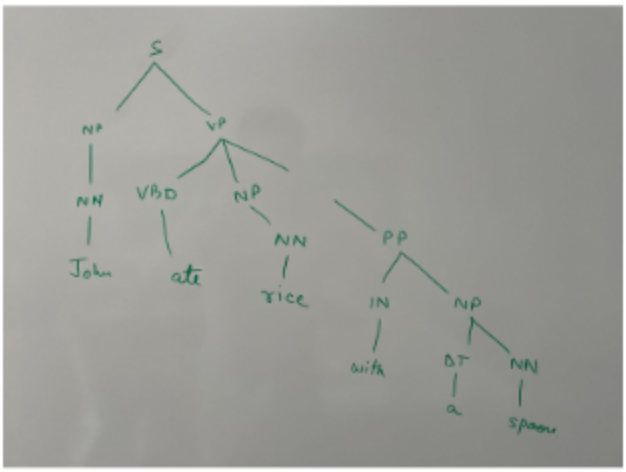
b. John ate rice with a spoon (2 marks)

S ⇒ **NP** VP ⇒ **NN** VP ⇒ John **VP** ⇒ John **VBD** NP PP ⇒ John ate **NP** PP ⇒ John ate **NN** PP ⇒ John ate rice **PP** ⇒ John ate rice **IN** NP ⇒ John ate rice with **NP** ⇒ John ate rice with **DT** NN ⇒ John ate rice with a **NN** ⇒ John ate rice with a spoon

OR

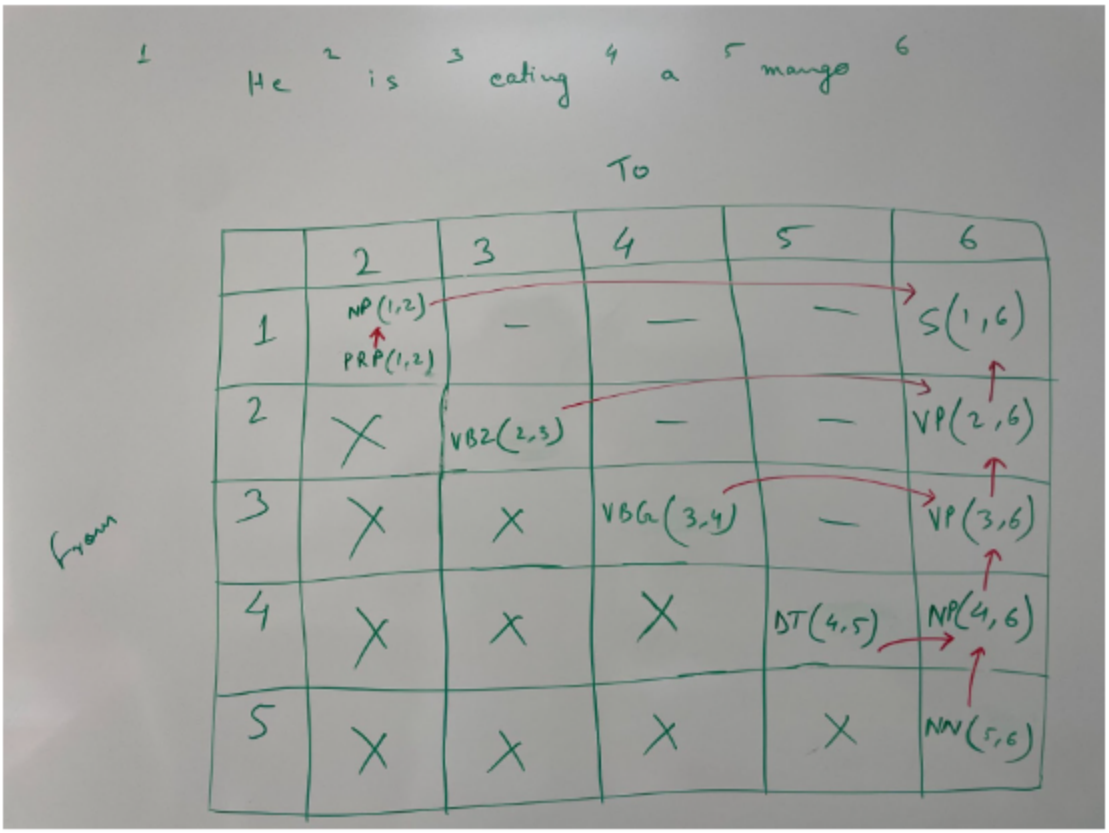
John ate rice with a spoon ⇒ **NN** ate rice with a spoon ⇒ NP **ate** rice with a spoon ⇒ NP VBD **rice** with a spoon ⇒ NP VBD **NN** with a spoon ⇒ NP VBD NP **with** a spoon ⇒ NP VBD NP IN **a** spoon ⇒ NP VBD NP IN DT **spoon** ⇒ NP VBD NP IN **DT NN** ⇒ NP VBD NP **IN NP** ⇒ NP **VBD NP PP** ⇒ **NP VP** ⇒ S

OR



4. Considering the following grammar, compute the CYK parsing table for the sentence "He is eating a mango". **[10]**

S → NP VP NP → PRP NP → DT NN VP → VBG NP VP → VBZ VP	PRP → He VBZ → is VBG → eating DT → a NN → mango



Note:

- Arrows and Indexes (from, to) with each grammar symbol are important and part of the parsing table. They are important to construct the parse tree in the next step of the dynamic programming.

Penalties:

- If you missed the arrows but indexes are there, an overall penalty of 4 marks.
- If you missed the indexes but arrows are there, an overall penalty of 4 marks.
- If you missed both regardless of the correct grammar symbols in the cell, you'll get zero.

5. Given a RNN-based sequence learning model, **[6]**

$$\text{RNN}(x_1, x_2, \ldots, x_n) \Rightarrow y_1, y_2, \ldots, y_m$$

mention the necessary conditions for the following learning paradigms

a. Sequence Transformation
   m >= 1 and n >= 1 (2 marks)

b. Sequence Labeling
   m >= 1 and n >= 1 and m == n (2 marks)

c. Sequence Classification
   m == 1 and n >= 1 (2 marks)

6. How are MEMM and CRF different from each other? MEMMs are trained locally to maximize the probability of each individual label, whereas CRF is trained globally to maximize the probability of the overall sequence.	**[2]**
7. Write the equations for computing the LSTM's output gate. Where, x_t, h_{t-1}, and c_t are the input, hidden state, and cell state, respectively. W_o is the learnable weight matrix. $o_t = \sigma(W_o \cdot [h_{t-1}, x_t])$ $h_t = o_t \otimes tanh(c_t)$ Note: Penalty of one mark for not defining variables of the equation.	**[2]**
8. What are the possible ways to model a span identification task? Provide the solution for the following sentence in the aspect-term-identification task: ***{Battery life}***$_{aspect\text{-}term}$ *is awesome but* ***{camera}***$_{aspect\text{-}term}$ *is dreadful.* We can model the tasks in one of following ways: ○ Sequence labeling ■ BIO encoding → Three classes *B*, *I*, and *O* (1 mark) ***Battery_B life_I*** *is_O awesome_O but_O* ***camera_B*** *is_O dreadful_O ._O* (1 mark) ■ Boundary detection – Identify the start and end indexes of the answer → Three classes *start*, *end*, and *none*. It will be a multilabel classification problem to handle the span of a single token. (1 mark) ***Battery_start life_end*** *is_none awesome_none but_none* ***camera_[start, end]*** *is_none dreadful_none ._none* (1 mark) ○ Generation (1 mark) ■ Generate the start and end indexes ***Battery life*** *is awesome but* ***camera*** *is dreadful. ⇒ <1, 2>, <6, 6>* (1 mark) **NOTE:** *If you have a different solution, we may ask you for a justification and will evaluate it according to its merit.*	**[6]**