National University of Computer & Emerging Sciences, Karachi
FAST School of Computing
AI Department
Fall 2023
Mid II Examination
8th November 2023 10:00 AM - 11:00 AM

Course Code: AI4001	**Course Name:** Fundamentals of Natural Language Processing
Instructor Name: Sumaiyah Zahid	
Student Roll No:	**Section No:**

Instructions:

- Return the question paper.
- Read each question completely before answering it. There are **5 questions and 2 pages.**
- In case of any ambiguity, you may make assumptions. But your assumption should not contradict any statement in the question paper.
- Show all steps clearly.

Time: 60 minutes. **Max Marks**: 30 points

Question 1 [10 Points (2.5 each)]: [CLO 2]

a) In a corpus of N documents, one document is randomly picked. The document contains a total of T terms and the term "data" appears K times. What is the correct value for the product of TF (term frequency) and IDF (inverse-document-frequency), if the term "data" appears in approximately one-third of the total documents?
formula for TF is K/T
formula for IDF is log(total docs / no of docs containing "data")
= log(1 / (⅓))
= log (3)
Hence correct choice is Klog(3)/T

b) You have trained word embeddings using a text dataset of m1 words. You are considering using these word embeddings for a language task, for which you have a separate labeled dataset of m2 words. Under which of these circumstances would you expect the word embeddings to be helpful? 1)m1 >> m2 or 2) m1 << m2. Explain your reasoning and give supporting examples.
m1 >> m2 (m1 is much larger than m2):
Rich Representation Learning: Larger datasets for training word embeddings allow for a more comprehensive understanding of word relationships, semantics, and contexts. Generalization: Word embeddings trained on a vast dataset tend to generalize better to unseen or smaller datasets.
Example: Wikipedia =m1

c) Imagine you're building a smart home system that uses natural language processing to understand and respond to voice commands. In this project, you plan to use Word2Vec (SkipGram) with a 1000-word vocabulary, mapping each word to 12-dimensional vectors for context understanding. Describe the dimensions of the individual word vectors, matrices and all the variables taken if the window size is 2.
x=[1000*1] Winput=[1000*12] h=[12*1] Woutput=[12*1000] y=[1000*1] // we will have four vectors of y

d) Word2Vec represents a family of embedding algorithms that are commonly used in a variety of contexts. Suppose in a recommender system for online shopping, we have information about co-purchase records for items x1, x2, . . . , xn (for example, item xi is commonly bought together with item xj). Explain how you would use ideas similar to Word2Vec to recommend similar items to users who have shown interest in any one of the items.

We can treat items that are copurchased with x to be in the ’context’ of item x. We can use those copurchase records to build item embeddings akin to Word2Vec. Then we can use a similarity metric such as finding the items with the largest cosine similarity to the average basket to determine item recommendations for users.

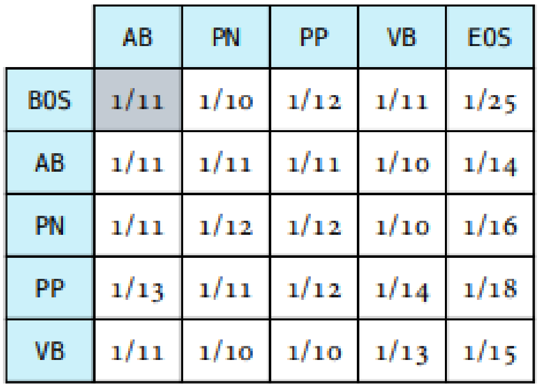
Question 2 [5 Points]: **[CLO 2]**

a) Give three examples of open vs closed words. **[1 Point]**
Open= noun, verbs, adjectives   Closed=i , he, she, on, at

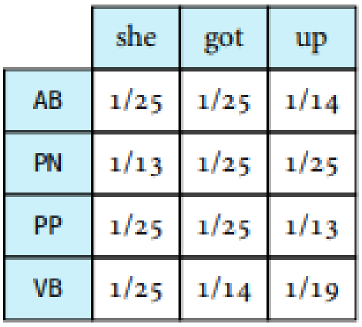
b) What is the major difference between CRF (Conditional Random Field) and HMM (Hidden Markov Model)? **[1 Point]**
HMM= Generative CRF=Discriminative

c) The following matrices specify (parts of) a hidden Markov model. The marked cell specifies the probability for the transition from BOS (Beginning of string) to AB. Which probability does this model assign to the following tagged sentence (word/tag)? she/PN got/VB up/AB. **[3 Points]**

1/10 *1/13*1/10*1*14*1/11*1/14*1/14= 0.0000255



	AB	PN	PP	VB	EOS
BOS	1/11	1/10	1/12	1/11	1/25
AB	1/11	1/11	1/11	1/10	1/14
PN	1/11	1/12	1/12	1/10	1/16
PP	1/13	1/11	1/12	1/14	1/18
VB	1/11	1/10	1/10	1/13	1/15



	she	got	up
AB	1/25	1/25	1/14
PN	1/13	1/25	1/25
PP	1/25	1/25	1/13
VB	1/25	1/14	1/19

Question 3 [5 Points]: **[CLO 2]**

You have been assigned a task with building a CRF-based NER model for Urdu language text. What types of features, such as word, contextual, lexical, and syntactic, would you consider, and why are they essential in improving the model's performance? What will be your strategy to generate features from words? Mention 5 feature functions with examples. What evaluation metric do you prefer and why?

Word Embeddings: Representing words as dense vectors capturing semantic meanings (e.g., pretrained Urdu word embeddings).
POS Tags: Assigning parts-of-speech to words (e.g., tagging verbs, nouns, or adjectives).
Prefix/Suffix Information: Capturing character-level features
Neighboring Word Context: Considering words before and after the current word as features for context
Gazetteer Matches: Identifying entities from domain-specific lists (e.g., recognizing location names or person names from a gazetteer).

Question 4 [7 Points]: **[CLO 3]**

a) Machine translation has the potential to break down language barriers and facilitate cross-cultural communication. However, it also poses risks in terms of misinformation and mistranslation. Explain the delicate balance between promoting accessibility and managing the risks in a globalized, machine translation-driven world. **[2 Points]**

Machine translation enables global communication but risks misinformation. Balancing accessibility requires improving accuracy, using human oversight, educating users, and adhering to ethical guidelines to manage errors and preserve cultural nuances.

b) Why is the brevity penalty in BLEU score constrained to be no larger than 1, and what is its significance in translation quality assessment? **[2 Points]**

The brevity penalty in BLEU is capped at 1 to prevent excessive penalization of shorter translations and to maintain fairness in assessing translation quality. This constraint ensures that shorter translations don't receive an unfairly harsh penalty, allowing for a balanced evaluation.

c) Calculate the BLEU score (limited to trigram) for the below machine translation: **[3 Points]**

Machine Translation: "The AI system demonstrated impressive capabilities."
Reference 1: "The AI system showed remarkable abilities."
Reference 2: "The artificial intelligence demonstrated remarkable skills."

Machine Translation n-grams:
Uni-gram: "The", "AI", "system", "demonstrated", "impressive", "capabilities"
Bi-gram: "The AI", "AI system", "system demonstrated", "demonstrated impressive", "impressive capabilities"
Tri-gram: "The AI system", "AI system demonstrated", "system demonstrated impressive", "demonstrated impressive capabilities"

Reference 1 n-grams:
Uni-gram: "The", "AI", "system", "showed", "remarkable", "abilities"
Bi-gram: "The AI", "AI system", "system showed", "showed remarkable", "remarkable abilities"
Tri-gram: "The AI system", "AI system showed", "system showed remarkable", "showed remarkable abilities"
Precision for uni-gram= 4/6
Precision for bi-gram= 2/5
Precision for Tri-gram= 1/4
BLEU score =e^0((4/6)^(1/3) *(2/5)^(1/3) *(1/4)^(1/3)) =0.4055
Reference 2 n-grams:
1-gram: "The", "artificial", "intelligence", "demonstrated", "remarkable", "skills"
2-gram: "The artificial", "artificial intelligence", "intelligence demonstrated", "demonstrated remarkable", "remarkable skills"
3-gram: "The artificial intelligence", "artificial intelligence demonstrated", "intelligence demonstrated remarkable", "demonstrated remarkable skills"
Precision for uni-gram= 2/6
Precision for bi-gram= 0/5
Precision for Tri-gram= 0/4
BLEU score =e^0((2/6)^(1/3) *(0)^(1/3) *(0)^(1/3)) =0

Question 5 [3 Points]: **[CLO 3]**

Briefly describe the problem of long range dependencies, and discuss how well each of the following architectures is able to deal with long range dependencies:

- Sliding window approach
- Simple Recurrent (Elman) Network
- Long Short Term Memory (LSTM)

Sliding Window Approach: Limited; can't effectively capture long-range dependencies due to fixed window size.

Simple Recurrent (Elman) Network: Limited; suffers from vanishing/exploding gradient issues, struggles with retaining information over long sequences.

Long Short-Term Memory (LSTM): Effective; designed to address vanishing gradient problem, can capture long-range dependencies by selectively retaining or forgetting information.

*****Best of luck while transforming Text into Machine Intelligence.*****