Roll No.												

PRESIDENCY UNIVERSITY

BENGALURU

Department of Research & Development

Mid - Term Examinations - SEPTEMBER 2024

Odd Semester: Ph.D. Course Work	**Date**: 28 /09/2024
Course Code: CSE900	**Time**: 2:00pm – 3:30pm
Course Name: Natural Language Processing	**Max Marks**: 50
Department: Computer Science & Engineering	**Weightage**: 25%

Instructions:

(i) Read all questions carefully and answer accordingly.

(ii) Do not write anything on the question paper other than roll number.

Part A

Answer ALL the Questions. Each question carries 5 marks. **4Qx5M=20M**

1 The formula for TF for a term t in a document d is given by: **5 Marks**

$$TF(t,d) = \log(count(t,d) + 1)$$

We consider the logarithm of the count to dampen the effect of document size. Explain why we add the **+1** at the end when we are taking the log. NOTE: Your answer should also specify what would happen if we ignored the +1 at the end.

2 Word vectors, such as GloVe, word2vec, etc. are created using the concept of ***distributional semantics***, as stated by John Firth "a word is characterized by the company it keeps". In other words, words which are used in the similar context have very high similarity. With the construction of word vectors using distributional semantics, we should expect synonyms (Eg. high and elevated) to have a high similarity, while antonyms (Eg. high and low) to have a low similarity. However, that is not the case. Using word2vec, we have the similarity between the synonyms high and elevated to be 0.460, while that of the antonyms high and low to be 0.742. Based on what you read above, and what you know of word vectors (word2vec, GloVe), and distributional semantics, explain why this happens. **5 Marks**

3 Define each of the following NLP tasks: **5 Marks**

a. Word tokenization
b. Part of speech tagging
c. Parsing

Arrange them in **increasing order of complexity**, from the **least to the most complex**

4	Differentiate between stemming and lemmatization. Use the following examples to demonstrate both stemming and lemmatization: a. Caring b. Helper c. Lovely d. Drove You need to provide BOTH a stem and a lemmatized form for each of the words.	**5 Marks**

Part B

Answer ALL Questions. Each question carries 15 marks.		**2QX15M=30M**
5	Python's NLTK library has a package called ***tokenize***. This package contains methods called ***sent_tokenize*** and ***word_tokenize***, which tokenize the text into lists of sentences and words respectively. Also, the python function len(x) will return the number of elements in the list x. Consider the following text data (text): "Long years ago, we made a tryst with destiny. Now the time has come when we shall redeem our pledge - not wholly or in full measure - but very substantially. At the stroke of the midnight hour, when the world sleeps, India will awake to life and freedom. A moment comes, but rarely in history, when we step out from the old to the new, when an age ends, and when the soul of a nation, long suppressed, finds utterance." Predict the output of the line print(len(sent_tokenize(text))). Here, text is the text data, and sent_tokenize(text) tokenizes the text into a list of sentences.	**15 Marks**
6	Consider that we use a very simple part-of-speech tag set, which has only 6 tags – noun (NN), verb (VB), adjective (JJ), adverb (RB), function word (FW), and punctuation mark (PM). The text in the previous question is tagged as follows by an expert tagger (the format of each token is of the form word_posTag: "Long_JJ years_NN ago_RB ,_PM we_FW made_VB a_FW tryst_NN with_FW destiny_NN ._PM Now_RB the_FW time_NN has_FW come_VB when_RB we_FW shall_FW redeem_VB our_FW pledge_NN -_PM not_FW wholly_RB or_FW in_FW full_JJ measure_NN -_PM but_FW very_RB substantially_RB ._PM At_FW the_FW stroke_NN of_FW the_FW midnight_NN hour_NN ,_PM when_FW the_FW world_NN sleeps_VB ,_PM India_NN will_FW awake_VB to_FW life_NN and_FW freedom_NN ._PM A_FW moment_NN comes_VB ,_PM but_FW rarely_RB in_FW history_NN ,_PM when_FW we_FW step_VB out_FW from_FW the_FW old_NN to_FW the_FW new_NN ,_PM when_FW an_FW age_NN ends_VB ,_PM and_FW when_FW the_FW soul_NN of_PM a_FW nation_NN ,_PM long_RB suppressed_VB ,_PM finds_VB utterance_NN ._PM" We have now built a part-of-speech tagger, which tags the text as follows:	**15 Marks**

	"Long_JJ years_NN ago_RB ,_PM we_FW made_VB a_FW tryst_NN with_FW destiny_NN ._PM Now_FW the_FW time_NN has_FW come_VB when_RB we_FW shall_RB redeem_VB our_FW pledge_NN -_PM not_RB wholly_RB or_FW in_FW full_JJ measure_NN -_PM but_FW very_RB substantially_RB ._PM At_FW the_FW stroke_NN of_FW the_FW midnight_JJ hour_NN ,_PM when_FW the_FW world_NN sleeps_VB ,_PM India_NN will_FW awake_VB to_FW life_NN and_FW freedom_NN ._PM A_FW moment_NN comes_VB ,_PM but_FW rarely_RB in_FW history_NN ,_PM when_FW we_FW step_VB out_FW from_FW the_FW old_JJ to_FW the_FW new_JJ ,_PM when_FW an_FW age_NN ends_VB ,_PM and_FW when_FW the_FW soul_NN of_PM a_FW nation_NN ,_PM long_RB suppressed_VB ,_PM finds_VB utterance_NN ._PM"	