# SMU MSDS 7337 Mid-Term Exam Summer 2020

# Short Essay Responses

(25 pts each, 300-500 words each).

1. Select one career or industry that makes use of applied NLP.
   1. Explain generally how that field or career utilizes NLP.
   2. Explain at least some methods of NLP that are very likely to be used in the career or industry you selected.
   3. Give at least one specific example of a use case for NLP within the chosen field, and explain how the problem or situation is (or could be) improved by applying NLP.
2. Choose one of the “trade-offs” in NLP that was covered in the asynchronous materials for this course.
   1. Explain the trade-off in general terms. Define the two choices.
   2. Explain the benefits and weaknesses of each side of the trade-off. Include at least one benefit and one weakness of each.
   3. Describe a work-situation that would make one of the choices in the trade-off much better, in terms of practical outcomes for you and your stakeholders on a project.

# NLP Networks

(50 pts).

I have a vocabulary of 10 words assigned the following indexes (in a dictionary):

{“the”: 0

“quick”: 1

“brown”: 2

“fox”: 3

“jumped”: 4

“over”: 5

“fence”: 6

“under”: 7

“car” : 8

“did”: 9 }

I have a network that classifies a sentence as a question or a statement. 0 means statement, 1 indicates a question.

I give you the following code as the network:

# truncate and pad input sequences

max\_sent\_length = 8

X\_train = sequence.pad\_sequences(X\_train, maxlen=max\_sent\_length)

X\_test = sequence.pad\_sequences(X\_test, maxlen=max\_sent\_length)

embedding\_vec\_length = 75

model = Sequential()

model.add(Embedding(top\_words, embedding\_vecor\_length, input\_length=max\_sent\_length))

model.add(LSTM(115, return\_sequences=True))

model.add(RNN(95))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])

Draw/Make a diagram of this network using an input sequence of “the car jumped over the fence”

Assumptions:

The sequence tokens are words, split by whitespace.

You may label a cell by its type—there is no need to show the inner connections of the LSTM cell. (A quick reminder—LSTM has 4 sets of gates/weights, but all those gates/weights have the same size matrix—that size is what I am after!)

1. Label each block and step by input/sequence step. Compute the dimensions of the weight for all steps. All inputs must be labeled by dimension. Include your original word ENCODING (notice not vector!) as input. You may omit bias
2. Write the initial vector form of the input sequence using only 1s and 0s
3. Find the average Glove Word Vector of your the input sequence (Spacy uses Glove vectors!)
4. Find the nearest word (in the above dictionary) to answer #3
5. What is the difference between the W(weight) matrix of the first LSTM sequence at time/sequence 0 and at time/sequence 5. How do you know this?
6. What is missing in the above code—something important is not determined and based on that, there are some minor adjustments or additions that need to be made Make a logical determination of what that missing piece of info should be based on the info given here and what additions or adjustments are necessary.

Example Diagram (Obviously you would fill in the “?”):

Dense Layer 1 (100 neurons), Wdim = (?, ?)

Dense Layer 2 (300 neurons), Wdim = (?, ?)

Classifier Dense Layer (300 neurons), Wdim = (?, ?)

Input, dimensions = (1 , 29)

(1,29)

(?,?)

(?,?)

(1,300)