UNIT 8 ASSIGNMENT

Natural Language Processing

## Instructions

The questions below will prepare you for future interviews as they relate to concepts discussed throughout the week. You’ve practiced these concepts in the coding activities, exercises and coding portion of the assignment. Now, let’s formulate your programming into well-thought responses.

Except as indicated, use this document to record all your assignment work and responses to any questions. At a minimum, you will need to turn in a digital copy of this document to your facilitator as part of your assignment completion. You may also have additional supporting documents that you will need to submit. Your facilitator will provide feedback to help you work through your findings.

**Note:** Though your work will only be seen by those grading the course and will not be used or shared outside the course, you should take care to obscure any information you feel might be of a sensitive or confidential nature.

*Begin your assignment by completing the questions below. Directions to submit your work can be found on the assignment page. Information about the grading rubric is available on any of the course assignment pages online. Do not hesitate to contact your facilitator if you have any questions about the assignment.*

Unit 8 Written Portion

# Implementing NLP Tasks

Answer the questions below about natural language processing.

## Questions:

1. What is NLP? What are real-world applications of NLP?

| NLP stands for Natural Language Problem. It is the class of ML and AI methods that deal with text as data. Some real-world applications of NLP are automated language translation, chatbots and topical classification of news articles. |
| --- |

1. Why and how do we have to transform raw text data for NLP tasks? Provide some examples of commonly used techniques in the NLP pipeline.

| Text needs to be converted to a structured numeric representation to be compatible with machine learning algorithms. We first clean and preprocess text by doing things such as removing stop words, performing lemmatization and creating tokens. We then create numerical feature vectors. Two approaches to do this are Vectorization, which maps a string token to a feature id and uses one of a few numeric representations. The other is Embedding, which maps a string token to a fixed length vector of numbers. |
| --- |

1. What is TF-IDF? And how is it calculated?

| TF-IDF stands for term-frequency, inverse-document-frequency. This is one method for converting text to a numeric representation. The TF component counts the frequency by which a word appears in a given example. The IDF is the inverse of the number of examples in which the word appears. The simplest variation is to multiple the above two components. There are variations however where the log of one or both components are used. |
| --- |

1. What is the difference between vectorizers and word embeddings?

| Vectorization maps a string token to a feature id and uses one of a few numeric representations. Embedding maps a string token to a fixed length vector of numbers. |
| --- |

1. What is the difference between a traditional neural network and a sequence-to sequence model? Why should sequence-to-sequence models be used in NLP? Explain the components of a sequence-to-sequence model.

| Feedforward neural networks are not ideal for text data because text data is inherently sequential and context-dependent. Often, a word itself is not meaningful until it's put into a sequence. Sequence-to-sequence models help solve this problem and are used with text data. They are used to handle different text applications, such as machine translation, text summarization, question answering, etc. Sequence-to-sequence models typically consist of two components, an encoder and a decoder.  An encoder is a neural network that takes in a sequence of words and outputs a vector or a code that can be viewed as a summary of the input sequence. A decoder is a neural network that takes in the vector output of an encoder and turns it into a scalar or sequence of outputs. These can be words represented by word embeddings or other things, depending on the application. |
| --- |

1. Compare and contrast a deep averaging network to a recurrent neural network.

| Both deep averaging networks and RNNs are both examples of sequence-to-sequence models that can be used for NLP tasks. Both have an encoder and a decoder. Deep averaging networks are arguably the simplest possible sequence-to-sequence model, where the encoder simply averages the word embeddings of the input sequence. Often this is too simple. Recurrent neural networks (RNNs) are neural networks that are applied over and over again at each element of a sequence to produce a summary vector (in the encoder) or an output sequence (in the decoder). Deep averaging ignore the ordering of the input text whereas RNNs do not. |
| --- |



*To submit this assignment, please refer to the instructions in the course*.