Natural Language Processing

James Wilkinson, Devyani Gauri, Josh Cheema, Lili Barsky, Kaleem Ahmed

Homework 1

David Demeter

January 23, 2022

Q.2 : We wrote a “_tag_sequence” function to covert the list of tokens (sequences), replacing the years, integers, dates, decimals and other numbers with their relevant tags. We chose to use Regular Expressions for this task. Within reg-ex we used the re.sub() function to replace the left most non-overlapping occurrences of the numerical pattern with the appropriate tag.

Q.3: To set up the training, validation and testing sets we followed a series of steps through our “generate_datasets” function. Our rationale behind the structure of the 80/10/10 split was based on our understanding that we should not split a single sentence between different datasets because we wanted each dataset to be as self-contained as possible, and to make sense as a stand-alone piece. We knew that sentences may occur often, so we tried to find the indices of the ideal 80/10/10 split and the “round-up” our indices to the end of the sentence. We started with an index of where the training data ends and where the validation data begins. We repeated the same for the end of the validation data and the beginning of the test data. We then calculated the number of tokens required to get to the end of the sentence for both the training and validation sets, and rounded up accordingly. We then set the corpus to the training set.

Q.4: For this task, we created a “threshold” function that takes a threshold value, parses the token list and then uses the <UNK> tag in place of the ones below threshold. We then created a two new lists, one with the tokens that occur at or above threshold, and one with tokens below threshold.

For our custom metrics, we chose:

1. Average length of words in training, validation and test data: We chose this metric because it approximates the distribution of the complexity of words

across our splits and shows how even the distribution. This can further lead to the investigation of how word length might impact further tasks performed on the corpus.

2. Number of tokens that appear in training but never appear in validation or test data: This metric shows how many tokens are used for training that the model does not test on later. These tokens are “wasted”, or at least provide a better depiction of how much the model might be memorizing from the training data in comparison to the validation and test data.