Capstone Project 2 Milestone Report

The Problem

Simply put, the problem/challenge that this project will attempt to 'solve' is that of *Question Answering*, an NLP discipline that is concerned with building systems that automatically answer questions posed by humans in a natural language. The goal of the project is to produce a closed-domain question answering system that (details on the dataset later), in an ideal case, will correctly answer answerable questions and not attempt to answer unanswerable questions. Furthermore, the project will serve as a personal challenge to develop in areas of Deep Learning and NLP.

Data Acquisition/Wrangling

The dataset is the '*Stanford Question Answering Dataset'* (SQuAD), it's a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable. It consists of 150,000 question-answer pairs, 100,000 of those questions are 'answerable' and 50,000 are 'unanswerable'. The aim of the dataset is to encourage further research and development in the field of Question Answering and NLP. More can be found at: https://rajpurkar.github.io/SQuAD-explorer/.

The data is in a JSON file so the acquisition process is as simple as downloading the file and reading in the file via pandas. The goal of the wrangling step is to transform the data from the JSON format into a pandas dataframe where each row is a question, and the columns are: the question, answer, context of the question, if the question is answerable etc.

After analysing the JSON it was clear a simple import was not possible, hence the need for nested for loops. We looped over the Wikipedia articles present in the dataset, and then over the 'contexts' (paragraphs in the Wikipedia article), extracting the questions/answers (along with other fields) for each context into a dataframe and appending the dataframes to a 'master dataframe'. We also add the 'context' and 'title' of the Wikipedia articles as columns. The 'answer_data' function was defined in response to the differences in JSON format for the answerable and unanswerable questions to extract the 'start_index' and 'text' fields for the answer. As mentioned above, simply put, the data wrangling steps taken above can be best summarised by going over each theme and each context in turn and extracting the relevant information to create rows of data in a dataframe, the details of the implementation are a result of the structure of the JSON file.