Week 10 Progress Report

Group Name: Natural Language Processing Engineer

Specialization: Natural Language Processing

Topic: Resume Extraction

Team Member Name: Ashraf Moumin

Email: ashrafmoumin1@gmail.com

Country: Turkey

University: Istanbul Technical University

Problem Description: Natural Language Processing is the application of computational techniques to the analysis and synthesis of natural language and speech. As such, this field is very important to a large number sectors, in particular those dealing with extensive bureaucratic processes. In particular, in this project, we want to work on automating resume information extraction. This problem is a text classification one in which the goal is to assign each resume to skillsets.

Data Understanding: The type of data needed is different than usual. The dataset we have is a .json file with a 'content' part consisting of text written by job applicants, labels about the text and some metadata. The actual information that is really needed is in the 'content' part of the resumes along with labels of skills of each applicant or other information to extract. As for the statistics of the dataset, the 'content' for each instance has on average around 3453 characters, which translates approximately to between 690 and 863 words (taking an average of 4 to 5 characters per word).

Modeling: I think embeddings are very important in these kind of text classification tasks. Now, I am thinking of a model taking the text content, using an embedding of different parts of the text sequentially, and leveraging a similarity algorithm with the embedding vectors of the skills and informations we want to detect and then taking all parts of the text with a high similarity score with the the type of information we have. For example, we can train a model to find an embedding for general informations we want to find (a vector for locations, university…). Note that these vectors for instance do not represent a specific university for example but that all universities have an embedding similar to it. As of now, I am using the sentence transformers library to get the embeddings.

Exploratory Data Analysis (EDA): The EDA was performed by statistically analyzing the distribution of word lengths and character length in the data. Moreover, a plot of some of the dimensions of embedded words was done, highlighting the proximity of semantically similar words. A heatmap showing the values of the dot product between different vectors as similarity score was also plotted.