Teknuance Internship Documentation

- Bhargav D

Aim:

To create an auto-modelling system which when given the Table names of a database will form clusters of the tables such that the Entity-Relationship Diagrams between the tables of a particular cluster can be formed using foreign key and primary key relationships.

Idea:

- Since no one has previously worked on the problem of clustering table names, I first started to search the internet for Research Papers and Project Examples where the people have worked on the clustering of similar words.
- I found that for clustering of the words we need to find the similarity between the words and for that there are many popular similarity measures that are commonly used such as Euclidean Distance, Manhattan Distance, MInkowski Distance, Cosine Similarity, Jaccard Similarity and Levenshtein distance.
- The way to use a Similarity Coefficient / Distance Metric as a metric to form clusters is to multiply those distances by -1 so as that similar words will have a greater value and dissimilar words will have lesser values.
- Using these values we can now apply various unsupervised clustering algorithms such as K-Nearest Neighbours, Hierarchical Clustering, Bag of Words(Dictionary Learning), etc.
- The main disadvantage of these methods is that we are supposed to pre-determine the number of clusters which is a major disadvantage in our problem statement.
- I applied KNN method on a list of words and then using the Elbow Method found the number of optimal clusters and the results were not that great.
- So then I further searched the Internet for Research Papers who have worked on clustering of Words and found about the Affinity Propagation Model, which is an Unsupervised Clustering Algorithm generally used in the case of Text based Clustering.

- I applied this algorithm on the sample list of words which I had and it performed way better than the KNN and the Hierarchical Clustering.
- So then I proceeded to try various similarity measures for the Algorithm and found that the Levenshtein distance performed the best and the in the Research Paper too they had used the same Levenshtein distance as the similarity measure while clustering.
- The next problem I faced was that the table names sometimes contain some common words such as ID, Details, Info, etc. at the end of the table name which affects the Performance of our Clustering Algorithm, so certain pre-processing of the table names is required to ensure this does not happen.
- So, I created a list containing all the last words of all the table names and found out the ones which occur more than 20% of the time and removed them from the table name before passing it to the clustering algorithm. This improved the performance of my clustering algorithm.
- The Plan for new words is use to either use the above similarity measure to keep a certain threshold pertaining to which we either join it in a cluster or form a new cluster for it, but I think when we encounter a new word, it would be better if we can run the model again after adding the new data point because the algorithm would do a better job at finding the appropriate cluster or forming a new cluster of it's own for the new data point.
- The next job was to create the Foreign Key Relationships between the tables of a cluster formed. The approach I was to use the primary key or part of the primary key of the first table as the foreign key of the other tables of the cluster.

Implementation:

All of the above ideas were implemented in Python and tested on Test Case Data either generated by me or taken from the internet.