*Natural Language Processing*

*Final Project Summary*

*by: Gal Aviezri 208782912 & Omri Shahar Revivo 315873471*

Our approach in this task was to use all the tools that we have learned throughout this course and the knowledge of which tool is better for what purpose.

**Embedding**

After examining multiple transformers, based on the conclusions from previous, we chose to use SBERT (NN based algorithm) with ‘all-MiniLM-L6-v2’ as the pretrained model to embed the sentences.

**Clustering**

for the given dataset, we wanted to create sub-groups with similar semantic meanings by transforming the requests to vectors and applying the best clustering algorithm in order to do so.

We tried 3 different clustering algorithms: DBSCAN, OPTICS and the Community-Detection based on Louvain's algorithm that is provided with sentence-transformers package.

OPTICS, the first candidate, produced coarse clusters and after examining numerous parameters we quickly concluded it was not the right tool for this task.

Secondly, DBSCAN; The best results were given when using eps=0.7 and min\_samples=5 as the parameters, along with a tuning pass over the results using a method named ‘fineTuneClusters’ that unclusters clusters which contains less than the threshold provided in the config.json.

Lastly, The CommDec algorithm, we searched for the optimal thresh in range [0.4,0.9], resulting in 0.65 to be the winner and min\_community\_size set to the value specified in the config json exactly, we were able to reach a finer granularity for each of the clusters resulting in the best performance.

**Extraction of cluster representatives**

In order to extract the most diverse representatives from a given cluster, we tried 3 different heuristics.

The first was

i.e. find the 3 vectors which maximize the sum of distances between them.

After trying that numerous times the results were Ok but not optimal.

Later we tried using the PCA algorithm and find 3 Principal Components - reducing the dimensions of our data to 3 and finding a vector that minimizes the distance to each base vector in the PCA domain.

Lastly, we had a eureka moment, remembering an algorithm we learnt in a previous course - “power iteration” with which we were able to find an orthogonal space, consists of the cluster’s component. Iterating over the eigenvalues, we then chose a single vector from an eigenvalue space, selecting the embedding that minimizes the distance to that eigenvector. That heuristic provided the best and most diverse outcome.

**Naming The Clusters**

Exploring several POS tokenization, and frequency techniques along with some linear algebra methods - we came up with an algorithm that takes an inspiration from the [median of medians](https://en.wikipedia.org/wiki/Median_of_medians) algorithm, and it got us the optimal results.

the algorithm:  
1. find the median embedding in the cluster.

2. break it to bigrams and trigrams (removing specific stop words)

3. return the median the n\_gram which represents best the median embedding (minimizes the distance from the median embedding)

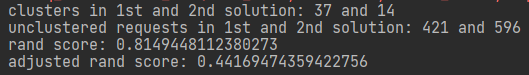
incase the suggested name is an existing name, we removed the median n\_gram and chose the new median after it.  
  
if again, the suggested name is an existing name, we assumed a semantic proximity between the clusters and united them.

Runtime: should not exceed 4 minutes for any dataset given for the task.

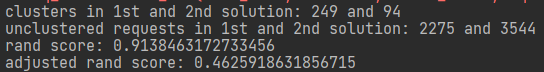
**Results:**

DBSCAN (initial approach for clustering)

dataset1:

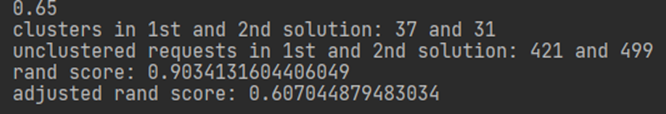


dataset2:



Sentence-Transformers.util.community-detection(Ultimate approach)

dataset1:



dataset2:

