NLP(DEEP) - Lab 07 - Theory Report

Eliot LECLAIR, Alex POIRON, Tom THIL, Aurelien VISENTIN

1 Explain the data structure. What are the three values returned by Beir, and how are they presented

We have 3 dictionnaries that are returned :

- **Corpus** : The corpus variable is the main data. We have for each data 3 values that represent the id of the document, a text in string format, and the title.
- **Queries** : The queries variable contains all the queries that are in a string format associated with a certain tag as key.
- **Qrels** : The qrel variable contains for a specific tag of a query, a dictionnary with as keys an id of a document and as values the degree of relevance for each following the related query. (0 = irrelevant, 1 or 2 = relevant)

2 Choice of the model

Among all the models available, we decided to choose the **msmarco-distilbert-base-v4**. The reasons are that simply because this model is tuned for cosine similarity in a first place. Secondly, because this specific model has the best result for the metric **NDCG@10**, which is the Normalized Discounted Cumulative Gain, and also the best result for **MRR@10** (Mean Reciprocal Rank). Moreover, its speed is quite correct for our lab. (GPU device from Google Colab).

3 Approximate nearest neighbours

3.1 Explain what the parameters you picked are, and why you chose them ?

In the function $\mathsf{init} - \mathsf{index}(\mathsf{max}_{\mathsf{elments}}, \mathsf{ef}_{\mathsf{construction}}, \mathsf{M})$ we have 2 hyperparameters that are $\mathsf{ef}_{\mathsf{construction}}$ and M.

For $\mathsf{ef}_{\mathsf{construction}}$, the parameter has the same meaning as ef, which is another hyperparameter. Ef corresponds to the size of the dynamic list for the nearest neighbors (used during the search). Higheref leads to more accurate but slower search.

Let's go back to $\mathsf{ef}_{\mathsf{construction}}$. This parameter controls also the $\mathsf{index}_{\mathsf{time}}/\mathsf{index}_{\mathsf{accuracy}}$. Bigger $\mathsf{ef}_{\mathsf{construction}}$ leads to longer construction, but better index quality. At some point, increasing $\mathsf{ef}_{\mathsf{construction}}$ does not improve the quality of the index. We chose $\mathsf{ef}_{\mathsf{construction}} = 500$.

For M, this is the number of bi-directional links created for every new element during construction. Reasonable range for M is 2-100. Higher M work better on datasets with high intrinsic dimensionality and/or high recall, while low M work better for datasets with low intrinsic dimensionality and/or low recalls.

Due to the fact that we work with high dimensional datasets, a M between 48 and 64 is consider as optimal performance. In our work, we chose $\mathsf{M} = 64$