

# INFORMATION RETRIEVAL (CS60092)

## Assignment 2,3 Report

Group 9  
Kushagra Chitkara

19EE10036

---

### Description of Tasks Completed

I worked on the retrieval model evaluation(2B) that was built in task 2A. To assess the model's performance, I calculated the model's Average Precision (AP) @10, AP@20, Normal Discounted Cumulative Gain (NDCG) @10, and NDCG @20 measures. This was repeated for each of the 35 inquiries. I then determined the Mean Average Precision (MAP) @10 and MAP@20 values by averaging the 35 AP values determined for each query in AP@10 and AP@20, respectively. The same method was utilised to calculate Average NDCG (averNDCG) @10 and averNDCG @20. The full procedure was carried out for Ranked Lists 1, 2, and 3.

### Design and Challenges Faced

Since a document could have several relevance ratings, I had to prioritise the rating with the highest iteration value. I programmed this by generating a dictionary and parsing all documents pertaining to a query. Each document's id functioned as the key, while its iteration value served as the key value. If I came across a document id with a higher iteration value, the dictionary was updated. If a lower value was noticed, I simply handed along the corresponding document ID. This dictionary is used to generate an array of relevant documents and to calculate the relevance of documents retrieved during DCG calculation.

For average precision, I created an array for relevant documents and another array for retrieved documents by reading the CSV files. I checked whether the retrieved document was in the array of relevant documents by parsing through the list of retrieved documents (10 or 20, depending on the metric). If true, the index for the retrieved document is saved in a distinct list. I could calculate the Precision@K values for each K1, K2, etc., and then find the average precision by averaging the Precision@K values. I repeated this procedure for each of the 35 queries to calculate the MAP value for a single ranked list.

I had to generate arrays for Discounted Cumulative Gain (DCG) and Ideal DCG values prior to computing NDCG (both 10 and 20) values. I iterated through the array of retrieved documents to obtain the DCG array and stored their relevance values using the earlier-created priority dictionary. This list is then sorted to achieve the ideal DCG values. The DCG and ideal DCG arrays were discounted by a factor of  $1/\log_2(\text{rank})$  before being added to prior indices to determine cumulative values. As python indexing begins at 0, it was tricky to keep the correct for loop arguments. After dividing the i-th value of the DCG list by the i-th value of the ideal DCG list, the computation for NDCG was pretty straightforward. The cumulative list's final value was given as NDCG. I repeated this method for each of the 35 searches to calculate the average averNDCG value for a single ranked list.

I then methodically saved the result values for each of the three ranked lists as text files. AP@10, AP@20, NDCG@10, NDCG@20, and all four average values were used. In task 3A, the MAP@20 and NDCG@20 metrics utilised the same code.