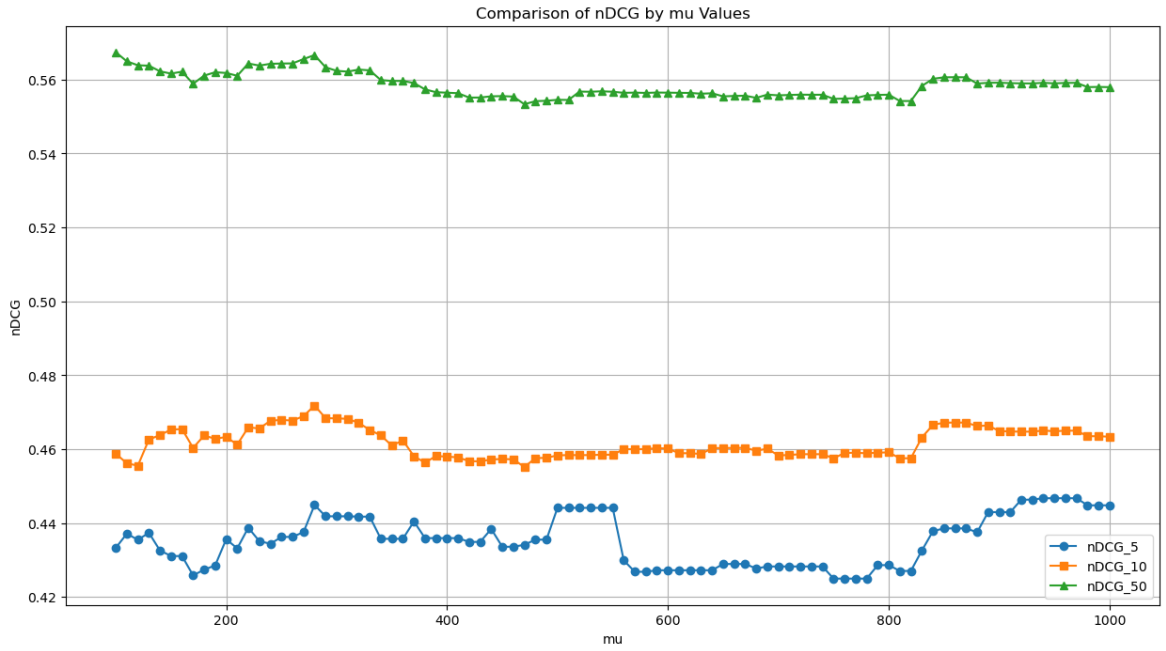
COL764 Assignment-2

Raj Chanda (2021ME10983)

Submitted on 04/10/2024

For Task 0, the first important parameter is mu – amount of smoothing used in Dirichlet Smoothing. Keeping other variables fixed, the optimal can be found which gives the highest nDCG.



From this plot, I have taken the value of mu for which the nDCG values are best.

mu = 280

Next, I have experimented with preprocessing techniques

- Remove non-ascii characters
- Remove non-English words
- Lemmatization and Stemming

Lower casing and numeric value removal and punctuation removal (based on results from Assignment 0) is always done for the preprocessing.

For all these experiments, standard NLTK libraries are used.

Parameter	nDCG_5	nDCG_10	nDCG_50
Removing non-ascii	0.43	0.45	0.53
Removing non-English	0.29	0.331	0.469

Non-ascii values are hence removed.

Parameter	nDCG_5	nDCG_10	nDCG_50
Lemmatization	0.423	0.451	0.576
Without Lemmatization	0.46	0.474	0.581

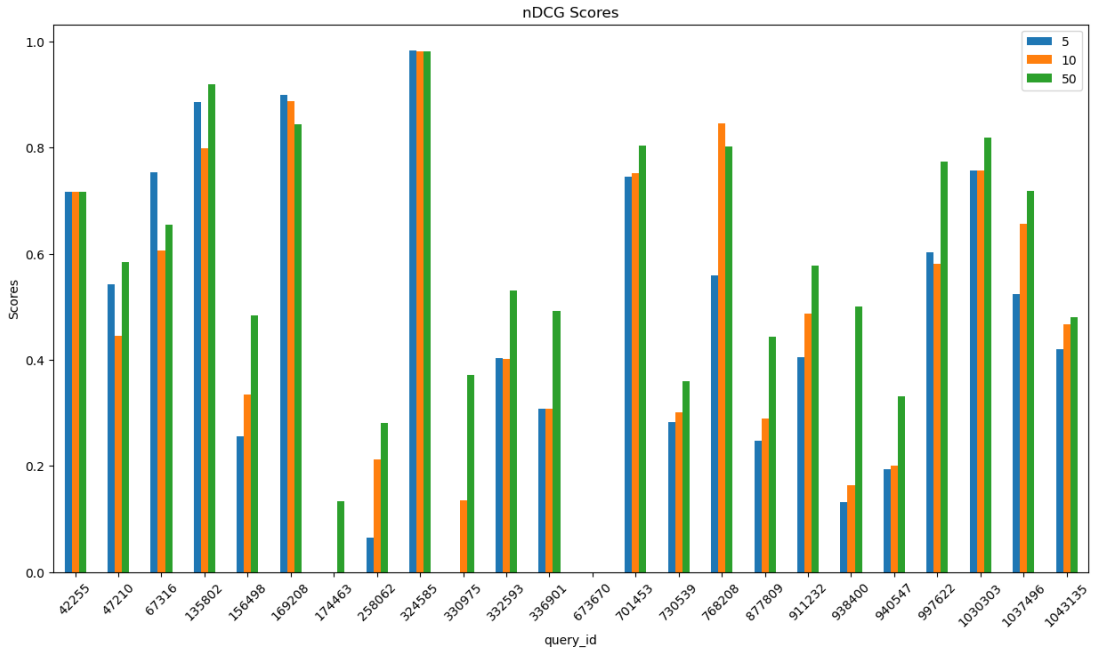
Lemmatization is hence not done

Final Set of Decisions and Parameters: -

Smoothing Constant (mu)	200
Lemmatization	No
Removing punctuations	Yes
Remove non-ascii characters	Yes
Stop word removal	Yes
Numeric Removal	Yes
Lower Casing	Yes

Task0 Metrics: -

	nDCG_5	nDCG_10	nDCG_50
Average Values	0.445	0.472	0.567



For Task 1 and 2: -

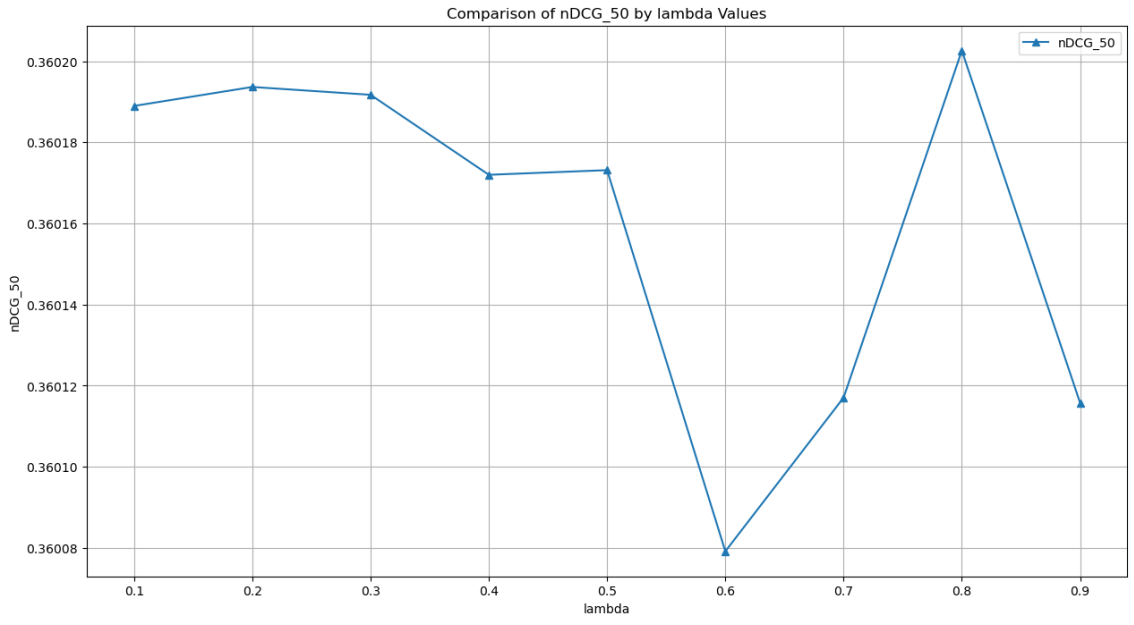
I first create relevance corpuses, from the top-K documents retrieved for each query. I use this corpus, to train the Word2Vec embeddings for each query. This gives a Embeddings matrix U of size V × d, where V is the size of the vocabulary and d is the dimensions of the embeddings trained. We then compute UU^T, which is a V × V matrix, where each term denotes the similarity between two vocabulary terms. Then I create the query vector q of dimension V × 1, where each elements denotes, how many times a vocabulary term is present in the query. We then compute UU^Tq which is a V ×1 vector denoting the proximity of each term in the vocabulary to the Query Terms. Pick the Top-K terms for this, and these are the Query Expansion terms. We use the weights of the previous multiplication and normalize them to get the Language Model corresponding to the expansion terms, terms which are closer to the query terms are given more weight. This model is called $\mathbf{p_q}^+$. We interpolate, the original query language model ($\mathbf{p_q}$), with this new model, to get the final query language model, according to the following equation.

$$p'_q = (1 - \lambda)p_q + \lambda p_{q+}$$

Using this query language model, I have computed the probability for generating a document, and then ranked the scores to get the reranking.

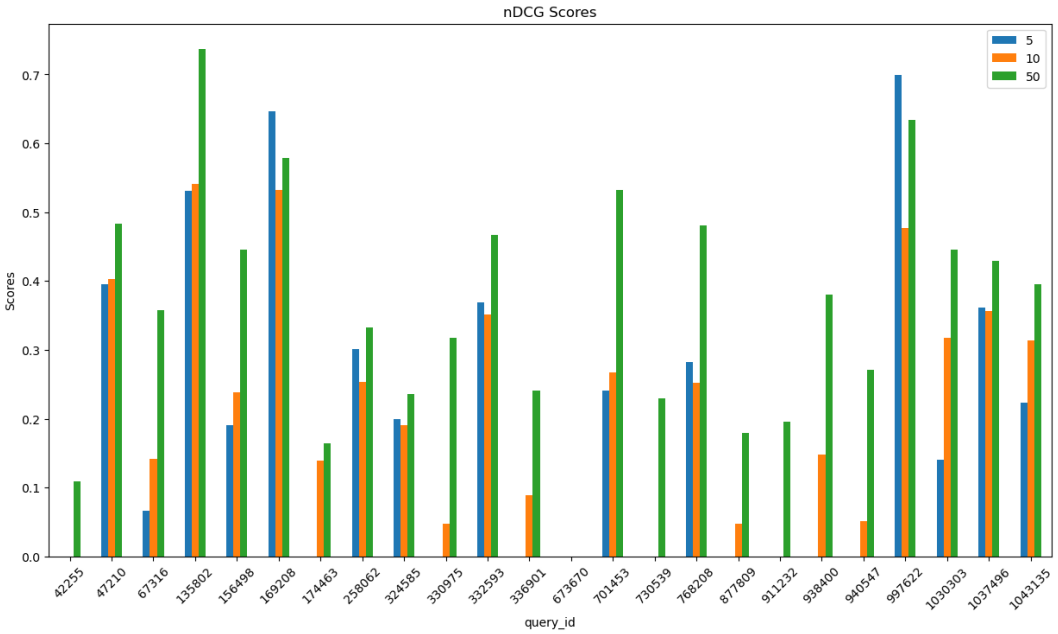
Number of expansion terms	10
Lambda	0.8
Vector size for w2v	100
Window size for w2v	5

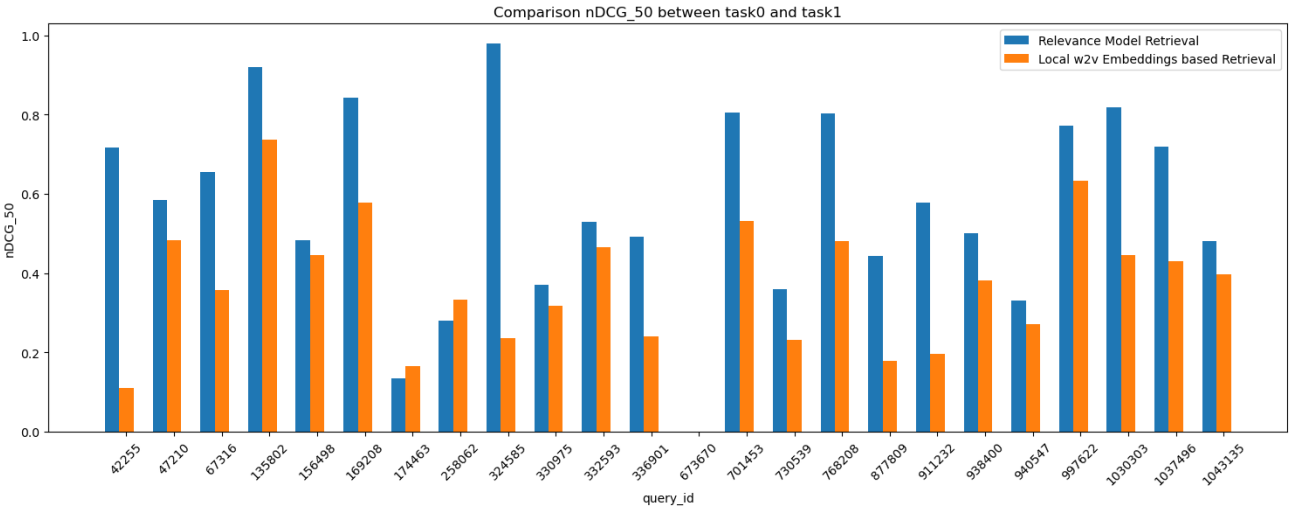
Word2vec was used from



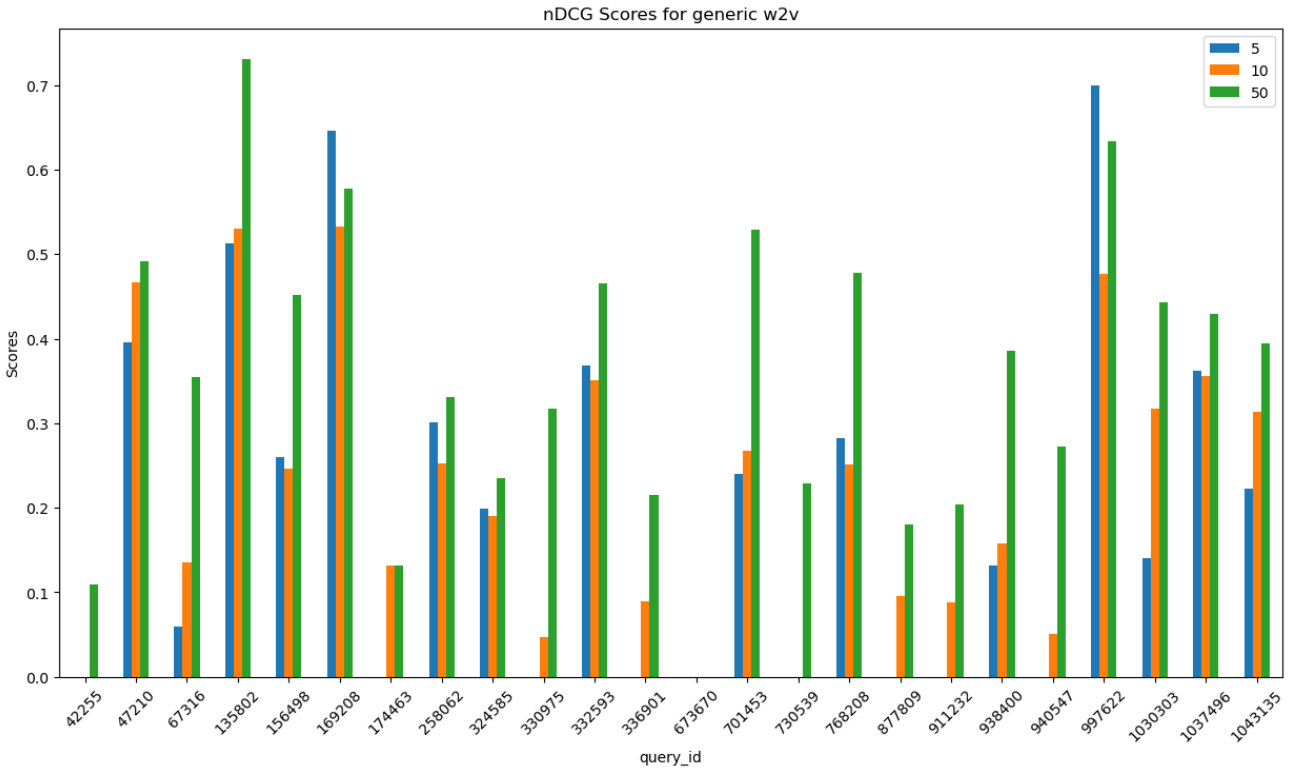
Task 1 Metrics

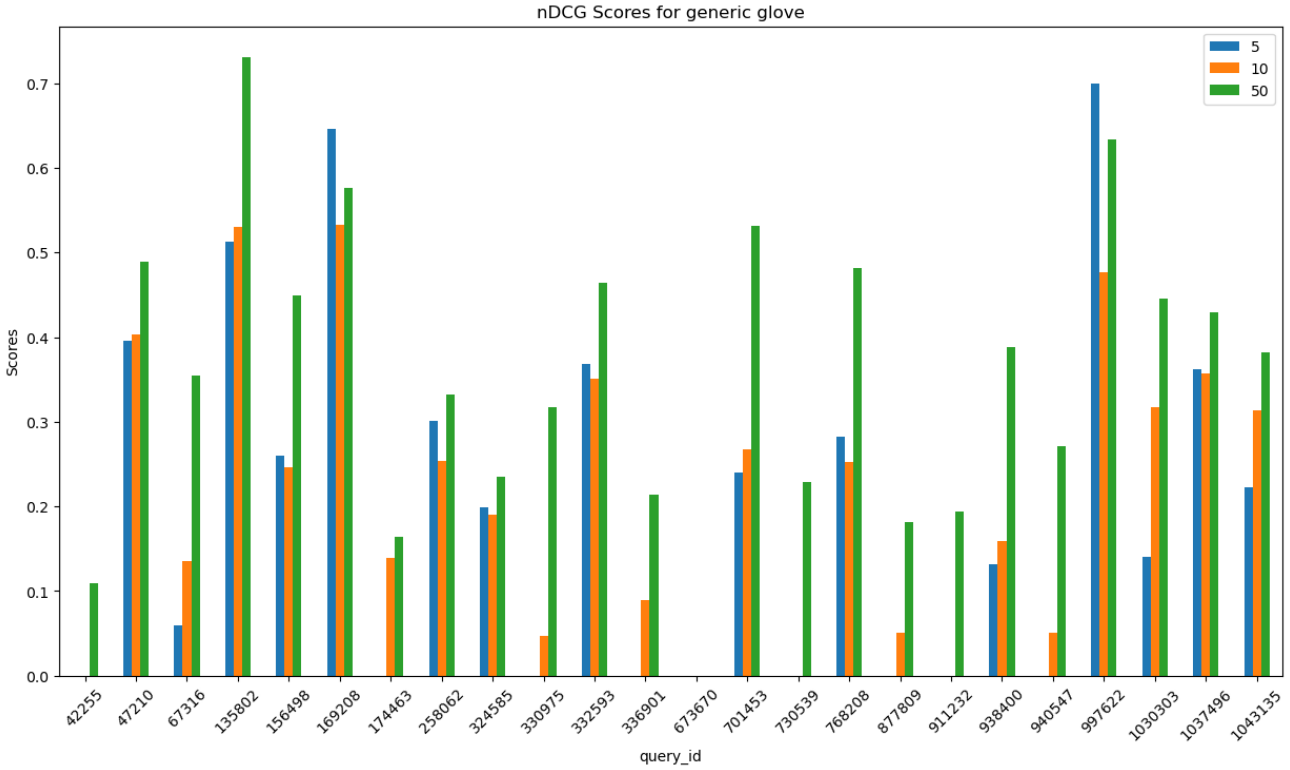
	nDCG_5	nDCG_10	nDCG_50
Average Values	0.194	0.215	0.361

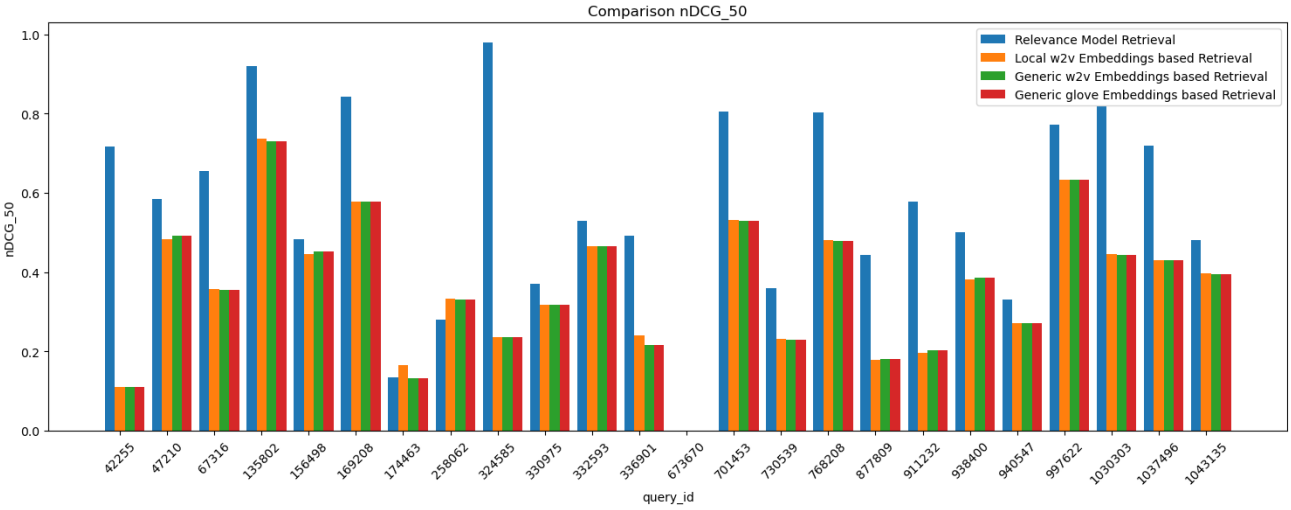




Task 2 Metrics






Comparison nDCG_50
Relevance Model Retrieval
Local w2v Embeddings based Retrieval
Generic w2v Embeddings based Retrieval
Generic glove Embeddings based Retrieval
nDCG_50
query_id
0.0
0.2
0.4
0.6
0.8
1.0
42255
47210
67316
135802
156498
169208
174463
258062
324585
330975
332593
336901
673670
701453
730539
768208
877809
911232
938400
940547
997622
1030303
1037496
1043135