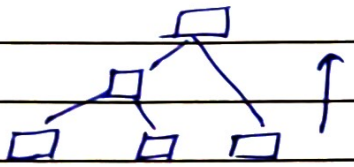
RAPTOR workflow

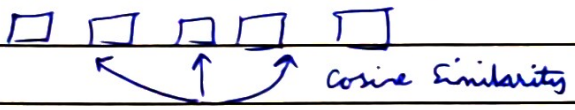
Goal: list of doc. splits → RAPTOR → even better top k splits
questions →

Steps:

1) Make pairs of available chunks using cosine similarity and use LLM to summarize them and form a chunk on their own, basically make a bottom up tree.



2) Squish the entire tree into 1 layer containing all the nodes.



multi-queries

3) Use cosine similarity to get the top-k nodes for our multi-queries with hopefully better context than the nodes we had before RAPTOR.

Note: Ensure generated nodes during the formation maintain the chunks' character limit.

The above RAPTOR method is used after clustering.

RAPTOR input nodes → (GMM+UMAP) → global clusters → (GMM+UMAP) → local clusters → bottom up tree

OG nodes of global clusters used to summarize it, cosine similarity performed on summaries and query to decide which clusters tree to traverse in order to save time.

The optimal number of clusters for the process are chosen using the BIC algorithm.

To calculate clusters a threshold is used and the responsibilities are extracted ~~from~~ by using GMM, using which soft clustering is performed.

Next, we get the summaries of all the clusters and perform cosine similarity search with our questions, this is done so that we don't end up using all the nodes to make the tree, resulting in very high computation costs.

After we select the clusters we form the bottom-up tree, flatten all summaries obtained into 1 list (collapsed tree) and perform similarity search to obtain the k-best nodes.

In the end the top k nodes are passed on to the LLM along with the question and the answer is obtained, hence bringing an end to the process.