Method 1: Hierarchical Search with Data Graph Plugins

- In traditional RAG models, the search engine retrieves a certain number of items from the database, which can lead to suboptimal results if the relevant information is scattered among many items.
- To solve this, we can use a hierarchical search method that uses data graph embedding.The idea is to represent the database as a graph, where entities, nodes, and relationships are nodes and edges, respectively.
- We can then learn to embed data graphs using techniques such as TransE or DistMulti, which capture semantic relationships between units and points.When searching, we can use these attachments to hierarchically group parts, to group semantically similar items.
- The RAG model can then take a set of clusters instead of individual points and use the cluster centroids as input to the generation module. This approach can help the model capture more nuanced relationships between units and points, leading to more accurate and informative responses.

Method 2: Competitive Training with Contrast Learning Another method to optimize the RAG model is to use adversarial training with contrast learning.

- The goal is to improve the model's ability to distinguish between important and irrelevant points and to provide more accurate and informative answers.
- We can achieve this by training a RAG model with a contrast loss function, where the model is represented by the set of positive and negative points for a given query.
- Positive points are those that are relevant to the study, while negative points are those that are not. During training, we can use adversarial techniques such as gradient inversion or adversarial examples to perturb the model's inputs and encourage it to learn stronger point representations.
- This can help the model better distinguish between important and irrelevant points and generate more accurate and informative answers.
- In addition, we can use techniques such as blending or cut blending to create improved versions of the points, which can help improve the robustness of the model to variations in the input data.
- By combining these two techniques, we can develop a more robust and accurate RAG model that is better equipped to handle complex questions and generate informative answers.