**Homework 2 Questions**

**PART A: Experimenting with Vector Store Query Design**

**A.I. Spreadsheet**

**A.II. Required:**

Explain your rationale for choosing the similarity metric you decided to use in the vector store. Please provide answers to the following questions in your homework submission:

1. ***What is one advantage of using the metric?***

Cosine similarity measures the angle between two vectors rather than their magnitude. This makes it highly effective for text-based embeddings because the direction of the vector (which represents semantic meaning) is more important than the absolute value of the embedding. Unlike the dot product, cosine similarity does not depend on the magnitude of the embeddings. This is crucial for text-based search, where different words and sentences might have embeddings of varying lengths, but their semantic similarity should be preserved.

1. ***What is one difference between using the metric you selected and the other similarity metrics? (e.g., cosine, dot product, and euclidean similarity metrics).***

Our selection: cosine

| Similarity metric | Vector properties considered |
| --- | --- |
| Cosine | Direction |
| Dot product | Direction & magnitude and normalized than cosine |
| Euclidean | Direction & magnitude |

**A.III. Spreadsheet**

**A.IV. Required:**

Qualitatively analyze the responses to your queries submitted to the vector store. Please provide answers to the following questions in your homework submission:

1. ***Did the queries retrieve the information you were expecting to obtain. Why or why not?***

The retrieved responses were mostly aligned with the expected information but had some limitations in relevance and precision.For the following examples:

For the query on the academic integrity policy, the second response provided a structured and direct policy definition, making it the most relevant. However, the first response contained irrelevant introductory content before reaching useful details on academic policies. This suggests a potential chunking issue, where extracted text includes unnecessary context.

For the query on policy violations for cheating, the second response was highly effective, listing specific violations such as unauthorized exam access, plagiarism, and falsified data. However, the first response was too broad, discussing general ethical behavior and citation requirements rather than explicitly defining "cheating." This indicates that the retrieval model may have ranked general integrity discussions higher than specific policy violations.

1. ***Why do you think the queries were successful / unsuccessful in retrieving the information you expected or needed?***

I would say that it was successful in retrieving the information but needs some improvements.

Success Factors:

* The best-ranked responses contained relevant information on academic integrity and violations.
* Key policies were retrieved, though sometimes mixed with broader ethical discussions.

Areas for Improvement:

* Chunking refinement: Remove unnecessary leading text to avoid irrelevant content.
* Better query expansion or ranking techniques: Ensure direct definitions of terms like "cheating" are prioritized over general policy discussions.

**B. Experimenting with the Vector Store Embeddings & Query Parameters**

**B.I. Required:**

Explain your rationale for selecting the query you choose in B.1. Please provide answers to the following questions in your homework submission:

1. ***Why did you choose this query vs. the other queries?***

We chose the query *"Where are pets allowed on CMU?"* because the response text it returned contained the most irrelevant information compared to the other queries. While the query was specifically about pet policies on campus, the retrieved text included unrelated details about bicycle impoundment policies, poster regulations, and quiet hours in residential areas.

This result highlights potential limitations in the vector store’s ability to retrieve highly relevant responses based on semantic similarity. It suggests that the similarity metric, chunking strategy, or embedding model used may not be optimally filtering for the most contextually appropriate sections. By analyzing this query’s output, we can better understand the effectiveness of the vector store’s retrieval process and explore possible improvements for future iterations.

**B.II. Spreadsheet**

**B.III. Spreadsheet**

**B.IV. Required:**

In observing the responses from the vector store to the queries created in B.1.: Please provide answers to the following questions in your homework submission:

1. ***Which ‘k’ parameter do you think retrieved the highest quality / best result?***

The optimal k parameter was k = 3 or k = 5, providing a balance of relevance and diversity. k=1 was too limited, often missing necessary context, while k=10 included irrelevant information.

1. ***Why do you think this parameter was the best to use with the query?***

The effectiveness of k depends on retrieval relevance and diversity:

* Relevance: Smaller k values return concise but potentially insufficient results. k=3 or k=5 ensures a broader yet relevant response.
* Diversity: Higher k values (e.g., k=10) introduce noise, making responses less useful. k=3 and k=5 provide enough variety without redundancy.

Thus, k=3 or k=5 was the optimal choice because it struck a balance between providing enough context without introducing excessive noise.

1. ***What would you recommend doing to improve the quality of the results?***
2. Preprocessing Responses:
   * Filter out redundant or irrelevant documents.
   * Apply semantic similarity scoring for better ranking.
3. Ranking Mechanism Improvements:
   * Prioritize contextually similar documents over purely vector-matched ones.
   * Use query expansion techniques to refine search results.

**B.V.**

In observing the responses from the vector store to the 6 queries created in B.2.: Please provide answers to the following questions in your homework submission:

1. ***Which ‘k’ parameter do you think retrieved the highest quality / best result?***

*\*\*We changed from Chunker: Text\_parser = SentenceSplitter(chunk\_size=1024) to Chunker: RecursiveCharacterTextSplitter(chunk\_size=1024, chunk\_overlap=100)\*\**

Based on the retrieved responses, k=5 provided the most balanced and relevant results. With k=3, the retrieved responses (Rows 49-51) contained a mix of unrelated information, such as bicycle impoundment policies and outdoor poster policies making it less effective at directly answering the query about pet policies. However, when k=5 was used (Rows 52-56), additional responses were included that covered actual pet policies, such as emotional support animals and their restrictions in housing, along with relevant details on student responsibilities regarding animals.

On the other hand, k=10 (Rows 57-66) introduced excessive noise, including details about unrelated policies, such as decorations in residence halls and vehicle transportation policy. These irrelevant responses diluted the effectiveness of the search, indicating that a higher k value may not always improve retrieval quality.

1. ***Why do you think this parameter was the best to use with the query?***

Using k=5 shows the best balance between coverage and relevance.

This was because:

* It retrieved more relevant pet policy details than k=3, which was too restrictive and often returned unrelated policies.
* Unlike k=10, it did not introduce excessive unrelated responses, such as those about poster regulations or alumni registration.
* Specifically, Row 53 provided an important sentence about student responsibilities when bringing pets, while Row 54 detailed emotional support animals and service animal policies in residence halls, making it a more comprehensive answer.

1. ***What would you recommend doing to improve the quality of the results?***

A possible improvement would be adjusting the character chunk size to ensure full policy sections are retrieved together rather than split across multiple chunks. Another improvement could be applying metadata filtering beforehand. For instance, the bicycle policy and poster policy were retrieved despite being irrelevant, so filtering results by document section titles could improve accuracy.