TIM 175: BUSINESS STRATEGY AND INFORMATION SYSTEMS
SPRING 2025

RAG Implementation

TIM 175 WEEK 7 LAB

In this week, you learned the fundamentals of vector databases and Retrieval Augmented Generation (RAG). Now, you will generate queries using a prompting workflow and interface with a pre-built vector database to retrieve relevant information. This will include an **individual deliverable (due Saturday at 11:59pm)** and a **team deliverable (due Monday 11:59pm)**

Readings:
See Week 7 Prelab or course spreadsheet.

Submission Link
Week 7 TIM 175 Submission Form (Spring 2025)

Brief Task Overview

- Setup your copies of this Google Spreadsheet and Google Colab
- Understanding and setting up the vector database
- Creating User Queries
- Converting User Queries to Query Objects
- Querying the Database
- Designing a Prompt to respond to users

You can use ChatGPT or other GenAI tools to inform any part of the assignment but: (1) you need to first form your own independent thoughts, (2) every word included in the submission needs to be something you've read, thought about, and decided to include, and (3) you should strive towards submitting the highest quality work you can rather than mediocre work that meets the requirements.

Project Overview

In this project, we want to take the YFIOB "What to be" transcripts and use them to create an experience where students can ask questions about their career journey and get responses grounded with actual advice and stories from professionals, and to provide a link to the podcast reference where the stories can be listened to further. Here is a description of the purpose of the project:

> ***To support*** *students in their career journey* ***trying to*** *obtain personalized advice and insights rooted in real-life stories and experiences from experienced professionals* ***because*** *of career-related stress and uncertainty, [****Project****]* ***will be a*** *simple app bringing together generative AI and YFIOB's What-To-Be podcast content that responds to student reflection about career anxieties and questions with support rooted in stories from the life journeys of career professionals in their community.* ***By doing so, it*** *helps students draw from the hard-earned wisdom of a large community of diverse professionals* ***instead of*** *leaving them to navigate their career journey on their own or with the limited mentors they have access to*

so that *all students have the opportunity to discover a fulfilling and successful career* ***rather than*** *being limited by their family or economic background.*

Setup:

Google Spreadsheet: Create a copy of this Google Spreadsheet Template: RAG results template. You will add rows for 5 user queries in the upcoming tasks and record the following under the respective columns by repeating Tasks 3-5 for each user query:

- The user query,
- Which podcast episode you know contains relevant content and what that content is (For 3 of the queries)
- The query object produced by your first prompt/workflow,
- The context string produced from the vector DB search,
- The output user response produced by your second prompt/workflow,

Google Colab: Make a copy of this Google Colab to edit. In the upcoming tasks, you will be using this Colab for:

- Setting up the database: You just need to add your API keys and run the cells.
- Writing a Prompt for creating Query objects: The code is all there you just need to add your prompt into it and run it
- Querying the database: The code has been provided for you. You just have to run the cells to retrieve the outputs.
- Writing a Prompt for creating user response. The code is all there you just need to add your prompt into it and run it

Change runtime type to T4 GPU. On your copy of the Google Colab, look at the top right corner you will see your connection status and a dropdown menu for "Additional connection options". From there change your runtime type to T4 GPU and save. GPU connection is preferred for AI tasks over CPUs primarily due to their parallel processing capabilities. This will make your notebook run more efficiently.

Task 1: Understanding and Setting up the Vector database

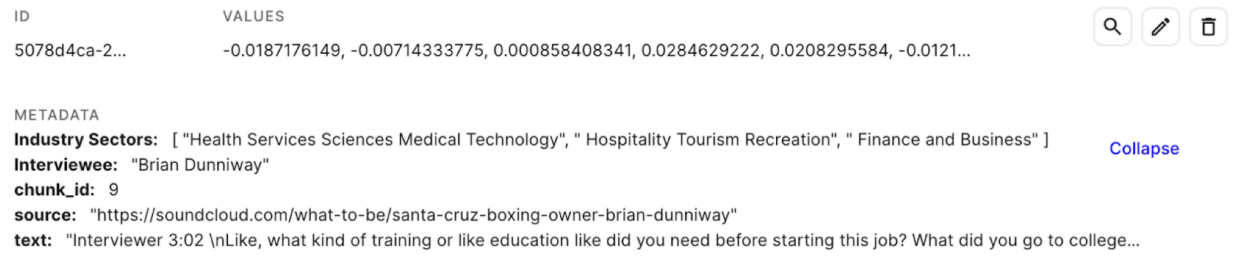
The first step is to understand the vector database and set it up using the code provided to you. Follow the instructions on the Google Colab to set up your database.

The database consists of Question-Answer chunks from the repository of cleaned transcripts from the podcast.

Formally, each item in the database includes:

- The Q-A segment
- The interviewee's name
- Industry sector tags (e.g., "Health Services, Sciences, Medical Technology")
- A SoundCloud source URL that links back to the original recording

Each entry in the vector database (we're using Pinecone in case you're curious) looks like the following:



The industry sector tags for each data item in the table will be drawn from the below list. Any data item without one of the 14 industry sector labels will be labeled with "Not categorized yet" and will also be included in the vector search even when industry sector filters are applied:

- Not categorized yet
- Architecture and Engineering
- Agriculture and Natural Resources
- Marketing, Sales, and Service
- Building, Trades, and Construction
- Energy, Environment, Utilities
- Fashion and Interior Design
- Manufacturing and Product Development
- Education, Child Development, Family Services
- Public and Government Services
- Finance and Business
- Arts, Media, and Entertainment
- Information and Computer Technologies
- Hospitality, Tourism, Recreation
- Health Services, Sciences, Medical Technology

Task 2: Creating User Queries

Your RAG-powered experience aims to support students' queries about career journeys by utilizing the stories and experiences from the What-To-Be podcasts. The students will be submitting queries by answering the following reflection prompt:

What would you like to ask industry professionals about their career journey? What stories would you like to hear them share? What stresses or concerns would you like them to speak to through their own experiences?

Task: Pick 2 transcripts that you are familiar with and are part of the database for testing retrieval in the next steps of this assignment.

Task: Define 5 diverse example user queries that might realistically be submitted by highschool students to the above question. Like in the past, you want them to be diverse so that you can stress test your prompts. For example, you might have questions that are directly asking for something specific, others that just express emotions and frustrations, questions that are short or long, etc.

Your 5 user queries should also be diverse in the following ways:

- Two queries should be defined based on content you know exists in the transcripts you chose above. In other words, you should know that there are high-quality stories or experiences that are relevant to the query so that when you're testing, you can know that if the RAG pipeline did not retrieve anything, there's something wrong.
- Another query should be defined based on content you know exists in the dataset, but should be based on a podcast transcript you are not yet familiar with. In other words, you should skim some other transcripts to find interesting content,
- For the last two queries, you do not have to define them based on known content. You should use these to enable more diversity in the questions themselves

You will one by one repeat Tasks 3-5 for each of the 5 queries. **Add your 5 queries and the episodes they link to to your copy of the RAG results spreadsheet.**

Task 3: Converting user queries into query objects:

The user queries you have designed above are in natural language, as would be when students submit their queries. To be able to query the database, we first need to convert these into query objects that can interact with the database.

Task: Design a prompt that takes user queries like the ones you defined above and outputs query objects that will query the vector DB to find relevant content.

The output of your prompt/workflow should be a query object with:

- the "**content_string_query**" property, which will be embedded for querying nearby documents,
- an optional "**industry_filter**" property, which can be used to filter documents on predefined industry sectors. If the user query is a general question that could draw insights from any industry sector, then do not include this property. If the user query is something where the

response really should be drawn from stories and experiences in a particular industry sector, then make sure to include this property.

Here is the format of a query object. Your response should only consist of this and nothing else. It cannot have any markdown or codefences like ```json ``` (you can instruct the LLM to do this in the prompt).

{
"content_string_query": [string that will be used to query the vector DB for passages close to the string within the embedding],
"industry_filter": [an optional property containing an array of one or more predefined industry sector tags you'd like to filter on],
}

Your prompt will need to take: 1) your user query, and 2) the complete list of industries (so that it knows what industry strings are allowable for use in the industry_filter property).

- Make sure the user queries you create actually read like things that a real student would type into an app.
- Make sure you use the predefined of industries listed at the end of Task 1

For example, suppose someone asked, "I feel stressed because all my friends know what they want to do, but honestly, I feel like I have no idea. Is that bad?" This is a pretty general question that could draw insights from professionals in any industry sector, so we do not want to have any industry filter.

An example of a good query object might be:

{
"content_string_query": "Advice and stories that can help with handling stress in career exploration. Specifically, to help someone who feels like they're the only one among their peers who doesn't have a clear idea what they want to do.",
}

Note that the "content_string_query" is not necessarily the same as the user query. Remember that the user query is what the user types into the app. The "content_string_query" is what you want to use to query the vector database, so it should represent potential topics that would be helpful in the response.

Add your prompt in the same Google Colab for Task 3 and run the cells to create the query object that will be used in the following tasks. Your Prompt should output the query object in the exact same format without any additional text because it needs to be used in Task 4 as it is - otherwise the code will not be able to run. If there is any additional text, make sure to clean it before moving on.

Remember to also Paste your Query Object in your copy of the spreadsheet.

Task 4: Querying the Database

We have already set up the Google Colab notebook to interface with the vector DB. *You only need to feed the query object you created from the previous step and run the cells.*

Task: The query object you return from your prompt above in Task 3 will be used to query the vector DB and will return to you a **string** containing all relevant information from up to 4 most relevant documents to your query (i.e. the closest 4 documents in the vector embedding out of those matching the filters you provided). The string that will be returned to you will look similar to the following (but with correct industry/source properties):

```
------------------------------------------------
'Top 4 Most Relevant Excerpts:'
------------------------------------------------
{'Passage': 'Interviewer  26:47\n'
            'Okay, what advice would you give to someone who is unsure about '
            'their career path?\n'
            '\n'
            'Pablo Orozco-Castro  26:53\n'
            'I would say for somebody who is unsure about their career path, '
            'what do you spend most of your time doing? Or, or, or thinking '
            'about? And I think from there, like you can identify, like, I '
            'could really do, I can really make a living doing what I love '
            "doing. And that's what I spend most of my time doing, or, or, or "
            'thinking about, I would say, like I said, tapping into resources, '
            'internally, or externally, like what Who do I have in my life, '
            'that could be a big support for me in this in this career path I '
            "would like to pursue. And because that's that support is also "
            "really important. There's also a there's also a really high "
            'burnout rate in in social work, basically, meaning that, like, '
            "social workers come in and out of the field, because it's so it's "
            "stressful work. And if you don't have that own your own social "
            "support, even going into a field that maybe you don't even know, "
            'or have any other sort of sites of support that can that can be '
            "stressful, that can be really stressful. It's not impossible, "
            "it's not impossible. And it's also important to utilize your "
            "social resources, maybe a friend that's in a field that you might "
            'be interested in, want to learn about that. So tapping into that. '
            "That's that's my advice.",
 'Interviewee': 'Pablo Orozco-Castro',
 'Source': '/content/yfiob_transcripts/110_PabloOrozco-Castro_WellnessCoordinatorEncompassCommunityServices.txt'}
**********
{'Passage': 'Interviewer  25:02\n'
            "Is there any sort of advice regarding? I don't know, finding a "
            "career? I've really I've no, I've said, I really loved what you "
```

In your google colab notebook, make sure your database is set up and then run the cells to obtain your relevant passages.

Copy and Paste the relevant passages output in your spreadsheet for your query.

Task 5: Design a prompt that responds to users.

This final step for the RAG pipeline is to design a prompt for taking the user query and the retrieved context to curate a user response to reply to the student.

Task: Design a prompt that responds to the user query by taking the user query and the retrieved contexts to produce the final response to the user. Use the prompt engineering techniques you have learned in this course to curate a strong prompt/workflow that will give a high quality response to the users query by using the retrieved context. Your response should:

- produce helpful answers to the questions students ask or to support them with advice, stories, or perspectives,
- be grounded in real experiences and insights from professionals contained in the What-To-Be podcast. Make these as substantive as possible. The value of the podcasts is that you have real rich stories from professionals. Tell those stories in your response,
- at the bottom / end of your response, it should direct the student to specific podcasts that the information is drawn from including links so that students can dig in further if they would like,
- the response should read naturally as a reply by a chatbot to the student query. The experiences and insights from professionals that you extract from the podcast should be integrated naturally into the response itself,

You will run the RAG pipeline for each of your 5 queries. You only need to replace the query in the last cell in the Google Collab and run it to run the full pipeline for you - you will not need to rerun all the cells again.
Add the user responses for each query into the spreadsheet.

Submission Instructions

Reflect and submit. Finally, reflect on the individual activity. Submit your responses to the questions directly on the Google Form along with your RAG Colab workflow link and spreadsheet. **IMPORTANT: MAKE SURE TO GIVE US PERMISSIONS TO ACCESS DOCUMENTS**

Team Assignment Instructions

In-section peer review of evaluation. Among the team members who are present at the section, conduct a peer review of each other's work. We don't have a strong preference for how you assign peer reviewers, but one way to do it is to use a cycle A → B → C → D → A. In other words, if you have 4 team members present (A, B, C, D), have A peer review B, B peer review C, C peer review D, and D peer review A.

Examining your team member's prompts and their spreadsheet, taking note of specific ways they could have improved in the prompts. Then:

- Write a two sentence reflection on what your team member did well in. For example, how did they write the prompts? Were there any prompt engineering techniques they used well? What was good about the quality of the responses generated by their prompt?
- Write a two sentence reflection on a concrete, specific way in which your partner could have improved their prompts and outputs. You should focus on what is most important / would make the biggest impact or improvement.

Note: The course staff will share some initial observations and pointers, but this is only based on a quick review of the individual submissions. You should take what they say into account, but make sure to be detail-oriented in thinking about the submission you are reviewing and how it should improve.

Compile your reviews. Create another copy of the RAG results spreadsheet for your team submission and compile the reviews from all of the participating team members in the 'Peer Reviews' tab. Specify for each review, who was the reviewer, who was being reviewed, the two sentence reflection on what your team member did well in, and the two sentence reflection on the most important way they could have improved.

We are looking at the quality of your reviews, so the way to maximize your points is to write the most helpful critique that points out the biggest way each team member can improve.

Iterate as a team. Work together to create a single final team version of the RAG pipeline, building on the work you have already done in the individual submission and considering what you might want to draw from each submission (e.g. specific user queries, prompts for creating query objects, prompt for creating user response.) Note: you do not have to use something from every individual submission. You should be trying to create a final RAG Pipeline workflow that produces the highest quality outcomes you can achieve.

Document your final pipeline for 15 diverse user queries. Choose a set of 15 diverse user queries from your individual submissions or create new user queries. At least 10 should be user queries for which you know there are high-quality stories or perspectives within the dataset. Run these through your new version of the RAG pipeline and document the results in your team version of the RAG Results Spreadsheet.

Reflect and submit. Submit your responses to the below questions directly on the Google Form along with your final prompt workflow and final spreadsheet. **IMPORTANT: MAKE SURE TO GIVE US PERMISSIONS TO ACCESS DOCUMENTS**

1. Write 3-4 sentences on your team's thought process. How did you create your final RAG Pipeline? What prompt engineering techniques did you use? What further improvements can be made? What feedback did you incorporate (from peer review or your tutor)?
2. Write a sentence or two describing the team dynamics. Were there any challenges you faced working in your team and how did you overcome them?
3. Please list each member of your team, whether they attended and engaged in section discussions, and their specific contributions.

Evaluation Rubrics

Individual submissions will be graded on a Check+, Check, Check-, Minus+, Minus scale according to the below rubric. The purpose of individual submissions is primarily to ensure that all members are contributing to their team, so graders will not be providing feedback on these.

Check+	Outstanding, one of the best in the class (102%),
Check	High quality, though not one of the best in minor ways (95%),
Check-	Completed the work, but needs significant improvement (80%),
Minus+	Low quality or missing significant portions (40%)
Minus	Did not do the work or barely did any work (0%)

Team submissions will be graded according to detailed rubrics to be posted on the rubric spreadsheet