Lab 6: RAG-enhanced Retrieval

6.1 Environment Setup

In [1]:

```
import os
from dotenv import load_dotenv

from sqlalchemy.engine.url import make_url

from IPython.display import Markdown
import textwrap

from langchain_core.prompts import PromptTemplate
from langchain_postgres.vectorstores import PGVector
from langchain_openai import OpenAIEmbeddings
from langchain_openai import ChatOpenAI
from langchain_core.output_parsers import StrOutputParser
```

In [2]:

```
# Load environment variables from .env file
load_dotenv()

connection_string = os.getenv("DB_CONNECTION")
openai_api_key = os.getenv("OPENAI_API_KEY")

# For this lab, we will use a shared table for demostration purpose.
shared_connection_string = make_url(connection_string).set(database="IST3

# Initialize the embedding model
embedding_model = OpenAIEmbeddings(model="text-embedding-3-large")

# Initialize the llm
llm = ChatOpenAI(temperature=0,
                 model="gpt-4o",
                 api_key=openai_api_key)
```

In [3]:

```
from sqlalchemy import create_engine, text

# Create the database engine
engine = create_engine(shared_connection_string)

# Test the connection by executing a simple query
try:
    with engine.connect() as connection:
        result = connection.execute(text("SELECT 1"))
        print("Connection Successful:", result.scalar())
except Exception as e:
    print("Connection Failed:", str(e))
```

```
Connection Successful: 1
```

In [4]:

```
# View the sample data from the langchain_pg_embedding table
try:
    with engine.connect() as connection:
        result = connection.execute(text("SELECT * FROM langchain_pg_embe
        rows = result.fetchall()
        print("Sample Data:", rows)
except Exception as e:
    print("Error accessing shared table:", str(e))
```

```
Sample Data: [('93928710-06b0-4ae8-9069-7b29cd2531be', UUID('41bd579a-1fd8
-403c-a1ac-216d6cc8d1f0'), '[0.016337192,-0.023811817,-0.01828898,-0.02145
5213,-0.0043553794,-0.029146707,-0.008848107,0.03163343,-0.012520361,0.052
134436,0.005186697,0.01192036 ... (38675 characters truncated) ... 34574,-
0.0008195704,-0.015166119,0.0007400531,0.008486665,-0.0026276854,-0.003791
5297,0.014486607,0.01189868,-0.0065384908,0.0053204303,-0.023262426]', 'Pr
eface\n“W hich of Peter Drucker’s books should I read?” “Where in your wor
k\ndo I find the best discussion on how to place people?” Not a week goes
... (73 characters truncated) ... four books\npublished over sixty-five ye
ars, even I find it difficult to answer these\nquestions.\nThe Daily Druck
er is intended to provide an answer', {'page': 9, 'year': '2004', 'title':
'The Daily Drucker: 366 Days of Insight and Motivation for Getting the Rig
ht Things Done', 'author': 'Drucker, Peter F.', 'source': 'The Daily Druck
er-2004.pdf'}), ('f5b95eba-f054-47cc-90f9-20538e08fc7f', UUID('41bd579a-1f
d8-403c-a1ac-216d6cc8d1f0'), '[-0.010645229,-0.016064199,-0.025607135,-0.0
15455239,-0.0030794854,-0.02969256,-0.007188035,0.030417144,-0.02138296,0.
024805466,0.025005884,0.0400217 ... (38702 characters truncated) ... 01067
6064,0.014753779,-0.021783793,-0.012071274,0.0013961747,0.0055191778,-0.01
003627,0.034471735,0.0029021932,-0.005657928,0.029846726,-0.005588553]',
'.\nThe Daily Drucker is intended to provide an answer. It presents in\nor
ganized form—and directly from my own writings—a key statement of\nmine, f
ol ... (171 characters truncated) ... onomy; a changing society;\ninnovati
on and entrepreneurship; decision making; the changing workforce;\nthe non
profits and their management; and so on', {'page': 9, 'year': '2004', 'tit
le': 'The Daily Drucker: 366 Days of Insight and Motivation for Getting th
e Right Things Done', 'author': 'Drucker, Peter F.', 'source': 'The Daily
Drucker-2004.pdf'}), ('75d34fb5-25d9-4d93-85c2-2cc8771778bf', UUID('41bd57
9a-1fd8-403c-a1ac-216d6cc8d1f0'), '[0.001684689,-0.03727856,-0.01865509,-
0.010671344,-0.024710089,-0.008766311,0.0037013753,0.003175713,-0.00525267
24,0.03882788,-0.024188379,0.05922200 ... (38632 characters truncated) ...
4267982,-0.0017746049,-0.013619796,0.004177633,0.014165221,-0.0011787883,
0.0006536198,0.026623026,0.011161435,-0.006446282,-0.006137999,-0.03908083
2]', '.\nBut the most important part of this book is the blank spaces at t
he bottom\nof its pages. They are what the readers will contribute, their
actions ... (109 characters truncated) ...  longtime friend and colleagu
e,\nProfessor Joseph A. Maciariello. It was his idea to bring together in
one\nvolume the best excerpts from my writings', {'page': 9, 'year': '200
4', 'title': 'The Daily Drucker: 366 Days of Insight and Motivation for Ge
tting the Right Things Done', 'author': 'Drucker, Peter F.', 'source': 'Th
e Daily Drucker-2004.pdf'}), ('8b55486b-1d71-4cd5-abb4-98fb90260228', UUID
('41bd579a-1fd8-403c-a1ac-216d6cc8d1f0'), '[0.018605381,-0.006636588,-0.02
1357486,-0.0065792524,-0.007360449,-0.04821918,-0.002830942,0.030617174,-
0.027951073,0.03440132,0.00468718,0.042055614 ... (38562 characters trunca
ted) ... .018691383,0.0041998276,-0.005106446,0.01553793,0.013258842,0.004
7982675,0.0038486477,0.024754616,0.013739027,-0.018806055,0.0025442643,-0.
045782425]', '. He then selected both the\nappropriate quotes and the comm
entaries on them from my books, scripts,\nand articles. The result is a tr
uly comprehensi ... (24 characters truncated) ... ffectiveness. My readers
and I owe a very great debt of gratitude to\nProfessor Maciariello.\n \nPE
TER F. DRUCKER\nClaremont, California\nSummer 2004', {'page': 9, 'year':
'2004', 'title': 'The Daily Drucker: 366 Days of Insight and Motivation fo
r Getting the Right Things Done', 'author': 'Drucker, Peter F.', 'source':
'The Daily Drucker-2004.pdf'}), ('a70e4474-2a6f-486a-8e1e-4f2a2da043fd', U
UID('41bd579a-1fd8-403c-a1ac-216d6cc8d1f0'), '[0.018273208,-0.016773168,-
0.020803576,-0.014712508,0.006966092,-0.027758306,-0.006814573,0.02534915
1,-0.027515875,0.039394975,0.030288674,0.04424358 ... (38690 characters tr
uncated) ... 22106642,0.0007571222,-0.016576193,0.0023580166,0.0089396285,
0.00828052,-0.0020777062,0.039152544,0.005269078,-0.014189767,0.006341075
5,-0.009644193]', 'Introduction\nI n putting together The Daily Drucker, I
have tried to distill and synthesize\nthe “tapestry” that Peter Drucker ha
s woven and continu ... (219 characters truncated) ...  capturing the esse
nce of the\ntopic. These proverbs, wise sayings, and quotes are mnemonic c
onstructs that\nremind one of the teaching on each topic', {'page': 11, 'y
ear': '2004', 'title': 'The Daily Drucker: 366 Days of Insight and Motivat
ion for Getting the Right Things Done', 'author': 'Drucker, Peter F.', 'so
urce': 'The Daily Drucker-2004.pdf'})]
```

In [5]:

```
# Establish a connection to the PostgreSQL vector store containing book d
book_data_vector_store = PGVector(
    embeddings = embedding_model,   # The embedding model used to generat
    collection_name = "Book_data",  # The name of the collection storing
    connection=shared_connection_string, # Database connection string def
    use_jsonb=True, # Enables JSONB storage format, optimizing metadata s
)
```

6.2 Query translation

In RAG, Query translation (also known as query transformation or query expansion) is a step where the user’s input query is reformatted, translated, or expanded to improve document retrieval from a vector database or search index. This step aims to enhance retrieval accuracy, especially in similarity-based searches, where slight variations in wording might lead to missed results.

The RAG summary image from this module's reading includes several query translation techniques:

1. Multi-Query Generation: Generate multiple variations of a query.
    - Given a single user query, an LLM rephrases it into multiple alternative queries.
    - The system then retrieves documents for each variation and merges the results.
2. RAG-Fusion: Retrieve documents using multiple retrieval strategies and fuse the best results.
    - Runs multiple retrieval strategies in parallel.
    - Merges results using re-ranking to select the most relevant ones.
3. Query Decomposition: Break down complex queries into simpler sub-queries.
    - A long or multi-part query is decomposed into smaller, independent queries.
    - The system retrieves documents for each sub-query and aggregates results.
4. Step-Back Prompting: Reformulate a high-level abstract question into a more concrete query for better retrieval.
    - If the user asks a vague or abstract question, the system "steps back" and rephrases it into a more answerable form.
    - Uses an LLM to generate a specific version of the abstract question.
5. Hypothetical Document Embeddings (HyDE): Instead of retrieving based on the query, the system generates a hypothetical answer and searches for similar documents.

- The LLM generates a "hypothetical" document as if it already had an answer.
- The hypothetical document is embedded into a vector representation.
- The system retrieves documents that are most similar to this generated text.

This lab demonstrates **Multi-Query Retrieval** technique based on LangChain's `MultiQueryRetriever`.

In [6]:

```
# Right click MultiQueryRetriever, select "Go to Definition", and then vi
# You don't need to run the code.
from langchain.retrievers.multi_query import MultiQueryRetriever
```

The LangChain `MultiQueryRetriever` implements the following prompt template to generate multiple variations of a query:

```
# Default prompt
DEFAULT_QUERY_PROMPT = PromptTemplate(
    input_variables=["question"],
    template="""You are an AI language model assistant. Your task
is
    to generate 3 different versions of the given user
    question to retrieve relevant documents from a vector
database.
    By generating multiple perspectives on the user question,
    your goal is to help the user overcome some of the
limitations
    of distance-based similarity search. Provide these
alternative
    questions separated by newlines.

    Original question: {question}"""
)
```

However, the `MultiQueryRetriever` has some limitations that impact its effectiveness in the RAG process.

1. No Built-in Reciprocal Rank Fusion (RRF), meaning it does not intelligently merge and rerank results from multiple query variations. Instead, results are retrieved for each query separately and concatenated only by removing duplicates. This can lead to redundancy and suboptimal ranking.
2. Does not support other types of retrieval such as Hybrid Retrieval, Metadata filtering, etc.

Thus, we will define our own multi-query prompt function for the RAG process instead of using `MultiQueryRetriever`.

In [7]:

```
# Create a prompt template for generating multiple variations of a user q
# This template rewords the original query into three different versions.
multi_query_prompt = PromptTemplate.from_template(
    """
    You are an AI language model assistant.
    Your task is to generate 3 different versions of the given user
    question to retrieve relevant documents from a vector database.
    By generating multiple perspectives on the user question,
    your goal is to help the user overcome some of the limitations
    of distance-based similarity search. Provide these alternative
    questions separated by newlines.

    Original user question: {question}""")

# Create a pipeline for generating multiple query variations.
# Step 1: The prompt template takes in the original user question.
# Step 2: The LLM generates alternative versions of the query.
# Step 3: The StrOutputParser ensures the output is treated as a string.
# Step 4: The lambda function splits the output into a list of queries, u
multi_query_generator = multi_query_prompt | llm | StrOutputParser() | (l
```

In [8]:

```
# Define an example user question.
question = "What did Drucker say about knowledge workers in the book?"

# Generate multiple queries using the multi_query_generator.
generated_queries = multi_query_generator.invoke({"question":question})

# Print a formatted output to display the generated queries.
# This helps visualize how the original question has been reworded.
print(f"\n{'='*40} List of generated queries for database search {'='*40}
print(f"Original question: {question}")

# Iterate over the list of generated queries and print each variation.
for i, query in enumerate(generated_queries):
    print(f"Generated query {i+1}: {query}")
```

```
======================================== List of generated queries for dat
abase search ========================================

Original question: What did Drucker say about knowledge workers in the boo
k?
Generated query 1: What insights did Drucker provide on knowledge workers
in his book?
Generated query 2: How does Drucker describe the role of knowledge workers
in his writings?
Generated query 3: What are Drucker's views on knowledge workers as discus
sed in his book?
```

6.3 Retrieval

In RAG, the Retrieval step is responsible for fetching relevant documents from an external knowledge source (e.g., a vector database, SQL database, or API) before generating an answer.

In addition to the basic similarity search based on raw similarity score, the RAG summary image from this module's reading shows several retrieval strategies to improve the quality of retrieved documents before passing them to the LLM. These include:

1. Re-Ranking: Sort documents based on relevance instead of similarity scores.

    - A query retrieves N most relevant documents from a vector store.
    - A re-ranking model assigns a new relevance score to each document.
    - The most relevant documents (top-k) are re-ranked and returned to the LLM.

2. RankGPT: Use an LLM-powered ranking model to evaluate and order retrieved documents based on their relevance to a given query.

    - A set of retrieved documents is passed to an LLM.
    - The LLM evaluates each document and assigns relevance scores.
    - The most relevant documents are selected for final output
3. RAG-Fusion: Use RFF to combine the rankings of documents retrieved from multiple variations of a user query.

    - Retrieve documents separately for each query.
    - RRF assigns scores to documents based on their rank in each individual retrieval.
    - RRF then fuses these scores to create a unified ranking.
    - Note: RFF can also be used to fuse mulitiple retrieval methods (e.g., BM25, FAISS)
4. Metadata Filtering: Filter documents based on metadata attributes (e.g., date, category, document type) before or after similarity search.

    - Pre-filtering approach uses metadata to arrow down the set of documents to be searched before performing the similarity search.
    - Post-filtering approach first retrieves documents based on similarity and then applies metadata filters to refine the results.
5. CRAG (Corrective RAG): Add a mechanism for error detection and correction within the RAG pipeline.

    - A retrieval evaluator assigns confidence scores to each document (e.g., correct, incorrect, or Ambiguous).
    - Define corrective actions for retrieval errors such as query expansion or refinement with relevancy of results.

6.3.1 RAG-Fusion

In this section, we will explore RAG-fusion using ** Reciprocal Rank Fusion (RRF)**, a simple yet effective method for combining ranked retrieval results. RRF is widely used in information retrieval to enhance ranking quality by aggregating multiple ranking lists.

RRF Formula

$$ \text{score} = \sum \frac{1}{\text{rank} + k} $$
where:

- **rank** represents the position of a document within an individual ranking list (starting from 1).
- **k** is a tuning parameter (typically set between 60-100) to prevent top-ranked documents from dominating the final ranking.

RRF ensures that lower-ranked results still contribute to the final ranking, making it particularly useful for balancing multiple retrieval strategies.

In [9]:

```python
from langchain.load import dumps, loads

# Define a function to apply RFF to a list of ranked retrieval results.
def reciprocal_rank_fusion(results: list[list], k=60):
    """
    Parameters:
    - results (list[list]): A list of ranked lists, where each sublist co
    - k (int): A tuning parameter that prevents top-ranked documents from

    Returns:
    - List[Tuple[dict, float]]: A list of tuples, where each tuple contai
        - The document (as a dictionary)
        - The fused score based on RRF
      The list is sorted in descending order of the fused scores.
    """

    fused_scores = {} # Dictionary to store the cumulative RRF scores for

    # Iterate through each ranked list of documents.
    for docs in results:
        for i, doc in enumerate(docs):
            # Convert the document to a string format (JSON) to use as a
            doc_str = dumps(doc)

            # Initialize the document's fused score if not already presen
            if doc_str not in fused_scores:
                fused_scores[doc_str] = 0

            # Compute the RRF score: 1 / (rank + k) with rank starting at
            rank = i + 1  # Adjust rank to start from 1 instead of 0
            fused_scores[doc_str] += 1 / (rank + k)

    # Sort the documents based on their fused scores in descending order
    reranked_results = [
        (loads(doc), score) # Convert JSON string back to its original do
        for doc, score in sorted(fused_scores.items(), key=lambda x: x[1]
    ]

    # Return the reranked list of documents with their fused scores
    return reranked_results
```

In [10]:

```python
# Define a retrieval chain for RAG fusion
retrieval_chain_rag_fusion = (
    multi_query_generator # Generate multiple queries
    | book_data_vector_store.as_retriever(
        search_type="similarity",   # Use vector similarity search to fin
        search_kwargs={'k': 5}      # Retrieve the top 5 most relevant do
        ).map()
    | reciprocal_rank_fusion    # Apply (RR) to merge and re-rank retriev
)
```

In [11]:

```python
# Execute the retrieval chain for RAG-Fusion and get the final reranked r
rag_fusion_results = retrieval_chain_rag_fusion.invoke({"question": quest

# Print a formatted header for displaying the retrieved results
```

```
print(f"\n{'='*40} List of reranked retrieved results {'='*40}\n")

# Display the total number of retrieved and reranked documents
print(f"Total number of results: {len(rag_fusion_results)}")

# Iterate through the retrieved documents and display them in a structure
for i, (doc, score) in enumerate(rag_fusion_results, start=1):
    # Display metadata: Source and page number
    display(Markdown(f"\n **From `{doc.metadata['source']}`, page {doc.me

    # Print the document content with proper text wrapping for better rea
    print(textwrap.fill(doc.page_content, width=100))

    # Add a separator for each document
    print("—" * 80)
```

```
======================================== List of reranked retrieved result
s ========================================

Total number of results: 9
```

```
C:\Users\yuyum\AppData\Local\Temp\ipykernel_47680\3315101807.py:35: LangCh
ainBetaWarning: The function `loads` is in beta. It is actively being work
ed on, so the API may change.
  (loads(doc), score) # Convert JSON string back to its original document
format
```

From `The Effective Executive-2002.pdf`, page 17

```
.  It takes his knowledge and uses it as the resource, the motivation, and
the vision of  other
knowledge workers. Knowledge workers are rarely in phase with each other,
precisely because they
are knowledge workers. Each has his own skill and his own  concerns. One m
an may be interested in
tax accounting or in bacteriology, or in training  and developing tomorro
w’s key administrators in
the city government
——————————————————————————————————————————————————————————————————————————
——————
```

From `The Essential Drucker-2008.pdf`, page 196

```
. Knowledge  workers, after all, first came into being in any substantial
numbers a generation ago.
(I coined the  term “knowledge worker” years ago.)   But also the shift fr
om manual workers who do
as they are being told—either by the task or by the  boss—to knowledge wor
kers who have to manage
themselves profoundly challenges social  structure
——————————————————————————————————————————————————————————————————————————
——————
```

From `The Daily Drucker-2004.pdf`, page 806

. Drucker analyzes the new realities of strategy, shows how to be a leader in periods of change, and explains the “New Information Revolution,” discussing the information an executive needs and the information an executive owes. He also examines knowledge-worker productivity, and shows that changes in the basic attitude of individuals and organizations, as well as structural changes in work itself, are needed for increased productivity. Finally, Drucker addresses the ultimate challenge of

---

From `The Daily Drucker-2004.pdf`, page 132

. They do not come with a merger or an acquisition. It is certain that the emergence of the knowledge worker will bring about fundamental changes in the very structure and nature of the economic system. ACTION POINT: What percentage of your workforce consists of people whose work requires advanced schooling? Tell these people you value their contributions and ask them to participate in decisions where their expertise is important. Make them feel like owners. Management Challenges for the 21st Century

---

From `The Effective Executive-2002.pdf`, page 68

." They produce ideas, information, concepts. The knowledge worker, moreover, is usually a specialist. In fact, he can, as a rule, be effective only if he has learned to do one thing very well; that is, if he has specialized. By itself, however, a specialty is a fragment and sterile. Its

---

From `The Daily Drucker-2004.pdf`, page 32

. Indeed, knowledge workers must know more about their areas than anyone else; they are paid to be knowledgeable in their fields. What this means is that once each knowledge worker has defined his or her own task and once the work has been appropriately restructured, each worker should be expected to work out his or her own course and to take responsibility for it. Knowledge workers should be asked to think through their own work plans and then to submit them

---

From `The Daily Drucker-2004.pdf`, page 321

```
23 May Knowledge-Worker Productivity Knowledge-worker productivity require
s that the knowledge
worker be both seen and treated as an asset rather than a cost. W ork on t
he productivity of the
knowledge worker has barely begun. But we already know a good many of the
answers. We also know the
challenges to which we do not yet know the answers. Six major factors dete
rmine knowledge-worker
productivity. 1. Knowledge-worker productivity demands that we ask the que
stion: “What is the task?”
2
──────────────────────────────────────────────────────────────────────────
──────
```

From `The Daily Drucker-2004.pdf`, page 323

```
. For knowledge workers are not programmed by the machine. They largely ar
e in control of their own
tasks and must be in control of their own tasks. For they, and only they,
own and control the most
expensive of the means of production—their education—and their most import
ant tool—their knowledge.
They do use other tools, of course, whether the nurse’s IV or the enginee
r’s computer. But their
knowledge decides how these tools are being used and for what
──────────────────────────────────────────────────────────────────────────
──────
```

From `The Daily Drucker-2004.pdf`, page 321

```
. 5. Productivity of the knowledge worker is not—at least not primarily—a
matter of the quantity of
output. Quality is at least as important. 6. Finally, knowledge-worker pro
ductivity requires that
the knowledge worker be both seen and treated as an “asset” rather than a
“cost.” It requires that
knowledge workers want to work for the organization in preference to all o
ther opportunities. ACTION
POINT: Apply steps one through five to your knowledge work. Management Cha
llenges for the 21st
Century
──────────────────────────────────────────────────────────────────────────
──────
```

6.3.2 RAG-fusion with Maximal Marginal Relevance (MMR)

Beyond basic retrieval, advanced search strategies such as **Maximal Marginal Relevance (MMR)** and **similarity score thresholding** can significantly impact the quality of retrieved results.

- MMR: Balances relevance and diversity. Instead of selecting only the most similar documents (which might be redundant), MMR diversifies the results by incorporating documents that provide new information while still being relevant. The λ parameter controls this balance: higher values (e.g. $\lambda = 1.0$) favor relevance, lower values ($\lambda = 0.0$) prioritize diversity, and $\lambda = 0.5$ represents a balanced approach.

- Similarity Score Thresholding: Filters out low-quality results that below a minimum relevance score.

In this section, we explore RAG-fusion with MMR.

In [12]:
```
# Define a retrieval chain for MMR with RAG-fusion
retrieval_chain_rag_fusion_mmr = (
    multi_query_generator
    | book_data_vector_store.as_retriever(
        search_type="mmr", # Use MMR retrieval to enhance diversity in re
        search_kwargs={
            'k': 5,         # Return the top 5 most relevant and diverse
            # Set `fetch_k` to be 2-5 times larger than `k` to give the M
            'fetch_k': 25,  # Initially retrieve 25 candidate documents b
            "lambda_mult": 0.5 # A balanced approach.
            }
            ).map() # fetch 25 documents for mmr with lambda_mult=0.5, bu
    | reciprocal_rank_fusion # rerank the documents using the reciprocal
)
```

In [13]:
```
# Execute the retrieval chain for RAG-Fusion with MMR and get the final r
rag_fusion_mmr_results = retrieval_chain_rag_fusion_mmr.invoke({"question

print(f"\n{'='*40} List of reranked retrieved results using MMR {'='*40}\

print(f"Total number of results: {len(rag_fusion_mmr_results)}")

for i, (doc, score) in enumerate(rag_fusion_mmr_results, start=1):
    display(Markdown(f"\n **From `{doc.metadata['source']}`, page {doc.me

    print(textwrap.fill(doc.page_content, width=100))

    print("-" * 80)
```

```
======================================== List of reranked retrieved result
s using MMR ========================================

Total number of results: 11
```

From `The Effective Executive-2002.pdf`, page 17

```
.  It takes his knowledge and uses it as the resource, the motivation, and
the vision of  other
knowledge workers. Knowledge workers are rarely in phase with each other,
precisely because they
are knowledge workers. Each has his own skill and his own  concerns. One m
an may be interested in
tax accounting or in bacteriology, or in training  and developing tomorro
w's key administrators in
the city government
--------------------------------------------------------------------------
------
```

From `The Daily Drucker-2004.pdf`, page 806

. Drucker analyzes the new realities of strategy, shows how to be a leader in periods of change, and explains the “New Information Revolution,” discussing the information an executive needs and the information an executive owes. He also examines knowledge-worker productivity, and shows that changes in the basic attitude of individuals and organizations, as well as structural changes in work itself, are needed for increased productivity. Finally, Drucker addresses the ultimate challenge of

---

From `The Daily Drucker-2004.pdf`, page 323

24 May Defining the Task in Knowledge Work In knowledge work, the how only comes after the what has been answered. I n manual work the task is always given. Wherever there still are domestic servants, the owner of the house tells them what to do. The machine or the assembly line programs the factory worker. But, in knowledge work, what to do becomes the first and decisive question. For knowledge workers are not programmed by the machine

---

From `The Daily Drucker-2004.pdf`, page 323

. They know what steps are most important and what methods need to be used to complete the tasks; and it is their knowledge that tells them what chores are unnecessary and should be eliminated. Work on knowledge-worker productivity therefore begins with asking the knowledge workers themselves: What is your task? What should it be? What should you be expected to contribute? and What hampers you in doing your task and should be eliminated? The how only comes after the what has been answered

---

From `The Daily Drucker-2004.pdf`, page 321

. 5. Productivity of the knowledge worker is not—at least not primarily—a matter of the quantity of output. Quality is at least as important. 6. Finally, knowledge-worker productivity requires that the knowledge worker be both seen and treated as an “asset” rather than a “cost.” It requires that knowledge workers want to work for the organization in preference to all other opportunities. ACTION POINT: Apply steps one through five to your knowledge work. Management Challenges for the 21st Century

---

From `The Daily Drucker-2004.pdf`, page 656

27 October Political Integration of Knowledge Workers Knowledge workers are, to coin a term,
“uniclass.” T he new majority, the “knowledge worker,” does not fit any interest- group definition.
Knowledge workers are neither farmers nor labor nor business; they are employees of organizations.
Yet they are not “proletarians” and do not feel “exploited” as a class. Collectively, they are
“capitalists” through their pension funds. Many of them are themselves bosses and have “subordinates

---

From `The Effective Executive-2002.pdf`, page 8

. The physician was the knowledge worker, with the nurse as his aide. In other words, up to recent
times, the major problem of organization was efficiency in the performance of the manual worker who
did what he had been told to do. Knowledge workers were not predominant in organization. In fact,
only a small fraction of the knowledge workers of earlier days were part of an organization. Most
of them worked by themselves as professionals, at best with a clerk

---

From `The Daily Drucker-2004.pdf`, page 367

. They have both mobility and self-confidence. This means they have to be treated and managed as
volunteers, in the same way as volunteers who work for not-for-profit organizations. The first thing
such people want to know is what the company is trying to do and where it is going. Next, they are
interested in personal achievement and personal responsibility—which means they have to be put in
the right job. Knowledge workers expect continuous learning and continuous training

---

From `The Daily Drucker-2004.pdf`, page 290

. Given this competitive struggle, a growing number of highly successful knowledge workers—business
managers, university teachers, museum directors, doctors —“plateau” in their forties. If their work
is all they have, they are in trouble. Knowledge workers therefore need to develop some serious
outside interest. ACTION POINT: Develop a serious satisfying outside interest. Managing in the
Next Society

From `The Essential Drucker-2008.pdf`, page 207

```
. The newly emerging dominant group are “knowledge workers.” Knowledge wor
kers  amount to a third or
more of the workforce in the United States, that is, to as large a proport
ion as  industrial blue-
collar workers ever were, except in wartime. The majority of knowledge wor
kers are  paid at least as
well as blue-collar workers ever were, or better. And the new jobs offer m
uch greater  opportunities
to the individual
──────────────────────────────────────────────────────────────────────────
──────
```

From `The Daily Drucker-2004.pdf`, page 796

```
.’ He is thoroughly at home in economics, political science, industrial ps
ychology, and industrial
sociology, and has succeeded admirably in harmonizing the findings of all
four disciplines and
applying them meaningfully to the practical problems of the ‘enterprise.’
” Drucker believes that
the interests of the worker, management, and corporation are reconcilable
with society. He advances
the idea of “the plant community” in which workers are encouraged to take
on more responsibility and
act like “managers
──────────────────────────────────────────────────────────────────────────
──────
```

6.3.3 Metadata Filtering

The section demonstrates metadata filtering, where a metadata filter is applied to the retrieval results before performing similarity search.

Metadata filters use comparison operators to refine search results based on structured attributes. Below are some commonly used operators:

- `$eq` : equal to
- `$ne` : not equal to
- `$gt` : greater than
- `$gte` : greater than or equal to
- `$lt` : less than
- `$lte` : less than or equal to

In [14]:

```python
# Define a retrieval chain that applies a metadata filter before similari
retrieval_chain_metadata_filter = (
    multi_query_generator
    | book_data_vector_store.as_retriever(
        search_type="similarity",
        search_kwargs={'k': 5,  # Retrieve only the top 5 most relevant d
                       'filter': {
                           'page': {'$lte': 300},    # Include only docume
                           'source': {'$eq' : 'The Daily Drucker-2004.pdf
                           }
                       }
                    ).map()
    | reciprocal_rank_fusion    # Apply RRF to merge and re-rank the filt
)
```

In [15]:

```
# Display the intermediate retrieval process
rag_fusion_metadata_filter_results = retrieval_chain_metadata_filter.invo

print(f"\n{'='*40} List of reranked retrieved results using metadata filt

print(f"Total number of results: {len(rag_fusion_metadata_filter_results)

for i, (doc, score) in enumerate(rag_fusion_metadata_filter_results, star
    display(Markdown(f"\n **From `{doc.metadata['source']}`, page {doc.me

    print(textwrap.fill(doc.page_content, width=100))

    print("—" * 80)
```

```
======================================== List of reranked retrieved result
s using metadata filter ========================================

Total number of results: 7
```

From `The Daily Drucker-2004.pdf`, page 132

```
. They do not come with a merger or an acquisition. It is certain that the
emergence of the
knowledge worker will bring about fundamental changes in the very structur
e and nature of the
economic system.   ACTION POINT: What percentage of your workforce consist
s of people whose work
requires advanced schooling? Tell these people you value their contributio
ns and ask them to
participate in decisions where their expertise is important. Make them fee
l like owners. Management
Challenges for the 21st Century
——————————————————————————————————————————————————————————————————————————
——————
```

From `The Daily Drucker-2004.pdf`, page 276

```
. The management of knowledge workers is a “marketing job.” And in marketi
ng one does not begin with
the question: “What do we want?” One begins with the questions: “What does
the other party want?
What are its values? What are its goals? What does it consider results?” W
hat motivates knowledge
workers is what motivates volunteers. Volunteers have to get more satisfac
tion from their work than
paid employees, precisely because they don’t get a paycheck. They need, ab
ove all, challenges
——————————————————————————————————————————————————————————————————————————
——————
```

From `The Daily Drucker-2004.pdf`, page 132

```
. For increasingly the ability of organizations to survive will come to de
pend on their “comparative
advantage” in making the knowledge worker productive. And the ability to a
ttract and hold the best
of the knowledge workers is the first and most fundamental precondition. W
hat does capitalism mean
when knowledge governs rather than money? And what do “free markets” mean
when knowledge workers are
the true assets? Knowledge workers can be neither bought nor sold
——————————————————————————————————————————————————————————————————————————
——————
```

From `The Daily Drucker-2004.pdf`, page 32

```
. Indeed, knowledge workers must know more about their areas than anyone e
lse; they are paid to be
knowledgeable in their fields. What this means is that once each knowledge
worker has defined his or
her own task and once the work has been appropriately restructured, each w
orker should be expected
to work out his or her own course and to take responsibility for it. Knowl
edge workers should be
asked to think through their own work plans and then to submit them
──────────────────────────────────────────────────────────────────────────
──────
```

From `The Daily Drucker-2004.pdf`, page 30

```
. What does this mean when the knowledge of the individual knowledge worke
r becomes an asset and, in
more and more cases, the main asset of an institution? What does this mean
for personnel policy?
What is needed to attract and to hold the highest-producing knowledge work
ers? What is needed to
increase their productivity and to convert their increased productivity in
to performance capacity
for the organization?   ACTION POINT: Attract and hold the highest-produci
ng knowledge workers by
──────────────────────────────────────────────────────────────────────────
──────
```

From `The Daily Drucker-2004.pdf`, page 290

```
. Given this competitive struggle, a growing number of highly successful k
nowledge workers—business
managers, university teachers, museum directors, doctors —“plateau” in the
ir forties. If their work
is all they have, they are in trouble. Knowledge workers therefore need to
develop some serious
outside interest.   ACTION POINT: Develop a serious satisfying outside int
erest. Managing in the
Next Society
──────────────────────────────────────────────────────────────────────────
──────
```

From `The Daily Drucker-2004.pdf`, page 286

```
. These people are as much manual workers as they are knowledge workers; i
n fact, they usually spend
far more time working with their hands than with their brains. So, knowled
ge does not eliminate
skill. On the contrary, knowledge is fast becoming the foundation for skil
l. We are using knowledge
more and more to enable people to acquire skills of a very advanced kind f
ast and successfully. Only
when knowledge is used as a foundation for skill does it become productive
──────────────────────────────────────────────────────────────────────────
──────
```

6.4 Generation

In RAG, the generation step is where the retrieved documents are used as context for an LLM to generate responses.

A basic generation simply uses retrieved documents from the previous step as context.

Limitations of basic generation include:

- The model may still hallucinate (generate incorrect information).
- If retrieval is incomplete or low quality, the generation may be misleading.
- The model does not verify facts—it simply generates based on provided documents.

Advanced generation techniques include:

- Self-RAG: Make the LLM self-aware by having it reflect on its own response, detect missing information, and refine the answer.
- RRR (Rephrase, Retrieve, Read): Iteratively rephrasing the query, optimizing retrieval, and structuring the response.

In [16]:

```
from operator import itemgetter

# Define a flexible Q&A prompt where the AI can use retrieved context as
# but is also allowed to incorporate its broader knowledge if necessary.
# This is useful when the retrieved documents may be incomplete or missin
flexible_QA_prompt = PromptTemplate.from_template(
    """You are a knowledgeable AI assistant answering the following quest
    If the provided context is relevant, incorporate it into your respons
    rely on your broader knowledge to provide the most accurate and infor

    **Question:** {question}

    **Reference Context (if applicable):**
    {context}

    **Answer:**"""
)

# Define a strict Q&A prompt where the AI must strictly adhere to the ret
# and avoid adding any external knowledge. If the context does not contai
# information, the AI should explicitly state that.
# This is ideal for high-accuracy applications such as legal, financial,
strict_QA_prompt = PromptTemplate.from_template(
    """You are an AI assistant tasked with answering the following questi
    based solely on the provided context. If the context does not contain

    **Question:** {question}

    **Context:**
    {context}

    **Answer:**"""
)


# Define the final RAG generation pipeline
final_rag_chain = (
    {"context": retrieval_chain_rag_fusion_mmr,  # Retrieve relevant docu
     "question": itemgetter("question")} # Extract the "question" field f
    | flexible_QA_prompt # Apply the flexible prompt as a demo
    | llm
    | StrOutputParser() # Convert the output into a simple string format
)

# Execute the final RAG pipeline
final_answer = final_rag_chain.invoke({"question": question})
```

In [17]:

```
# Display the final AI-generated answer based on retrieved context
print("\n" + "="*60 + " Final Answer " + "="*60 + "\n")
display(Markdown(f">\n> {textwrap.fill(final_answer, width=100)}"))

# Display the list of generated query variations for database search
print(f"\n{'='*40} List of Queries for Database Search {'='*40}\n")

# Show the original user question before query transformations
print(f"**Original Question:** {question}")

# Iterate over the generated queries and display them sequentially
for i, query in enumerate(generated_queries, start=1):
    print(f"**Generated Query {i}:** {query}")

# Display the list of retrieved documents
print(f"\n{'='*40} List of Reranked Retrieved Results Using Metadata Filt

# Print the total number of retrieved and re-ranked results
print(f"Total Number of Retrieved Results: {len(rag_fusion_mmr_results)}\

# Iterate through the retrieved documents and display their metadata and
for i, (doc, score) in enumerate(rag_fusion_mmr_results, start=1):
    display(Markdown(f"**Result {i}:**\n**From `{doc.metadata['source']}`

    print(textwrap.fill(doc.page_content, width=100))

    print("-" * 80)
```

```
============================================================ Final Answer
============================================================
```

> Peter Drucker, in his writings, particularly in "The Effective Executive" and "The Daily Drucker,"

emphasized the unique nature and importance of knowledge workers. He described knowledge workers as individuals who use their knowledge as a resource, motivation, and vision, often working independently with their own skills and concerns. Unlike manual workers, knowledge workers are not programmed by machines or assembly lines; instead, they must determine what tasks to undertake, making the "what" of their work a critical question before the "how." Drucker also highlighted that knowledge workers are rarely in sync with each other due to their specialized skills and interests. He noted that their productivity is not primarily about the quantity of output but the quality, and they should be seen and treated as assets rather than costs. Furthermore, Drucker pointed out that knowledge workers need to be managed as volunteers, emphasizing personal achievement, responsibility, continuous learning, and training. Overall, Drucker believed that the productivity of knowledge workers requires changes in attitudes and organizational structures, and that they should be integrated politically as a "uniclass," not fitting traditional interest-group definitions.

```
======================================== List of Queries for Database Sear
ch ========================================

**Original Question:** What did Drucker say about knowledge workers in the
book?
**Generated Query 1:** What insights did Drucker provide on knowledge work
ers in his book?
**Generated Query 2:** How does Drucker describe the role of knowledge wor
kers in his writings?
**Generated Query 3:** What are Drucker's views on knowledge workers as di
scussed in his book?

======================================== List of Reranked Retrieved Result
s Using Metadata Filter ========================================

Total Number of Retrieved Results: 11
```

Result 1: From `The Effective Executive-2002.pdf`, Page 17

```
.  It takes his knowledge and uses it as the resource, the motivation, and
the vision of  other
knowledge workers. Knowledge workers are rarely in phase with each other,
precisely because they
are knowledge workers. Each has his own skill and his own  concerns. One m
an may be interested in
tax accounting or in bacteriology, or in training  and developing tomorro
w's key administrators in
the city government
```

Result 2: From `The Daily Drucker-2004.pdf`, Page 806

```
. Drucker analyzes the new realities of strategy, shows how to be a leader in periods of change, and
explains the “New Information Revolution,” discussing the information an executive needs and the
information an executive owes. He also examines knowledge-worker productivity, and shows that
changes in the basic attitude of individuals and organizations, as well as structural changes in
work itself, are needed for increased productivity. Finally, Drucker addresses the ultimate
challenge of
```

Result 3: From `The Daily Drucker-2004.pdf`, Page 323

```
24 May Defining the Task in Knowledge Work In knowledge work, the how only comes after the what has
been answered. I n manual work the task is always given. Wherever there still are domestic servants,
the owner of the house tells them what to do. The machine or the assembly line programs the factory
worker. But, in knowledge work, what to do becomes the first and decisive question. For knowledge
workers are not programmed by the machine
```

Result 4: From `The Daily Drucker-2004.pdf`, Page 323

```
. They know what steps are most important and what methods need to be used to complete the tasks;
and it is their knowledge that tells them what chores are unnecessary and should be eliminated. Work
on knowledge-worker productivity therefore begins with asking the knowledge workers themselves: What
is your task? What should it be? What should you be expected to contribute? and What hampers you in
doing your task and should be eliminated? The how only comes after the what has been answered
```

Result 5: From `The Daily Drucker-2004.pdf`, Page 321

```
. 5. Productivity of the knowledge worker is not—at least not primarily—a matter of the quantity of
output. Quality is at least as important. 6. Finally, knowledge-worker productivity requires that
the knowledge worker be both seen and treated as an “asset” rather than a “cost.” It requires that
knowledge workers want to work for the organization in preference to all other opportunities. ACTION
POINT: Apply steps one through five to your knowledge work. Management Challenges for the 21st
Century
```

Result 6: From `The Daily Drucker-2004.pdf`, Page 656

27 October Political Integration of Knowledge Workers Knowledge workers are, to coin a term,
“uniclass.” T he new majority, the “knowledge worker,” does not fit any interest- group definition.
Knowledge workers are neither farmers nor labor nor business; they are employees of organizations.
Yet they are not “proletarians” and do not feel “exploited” as a class. Collectively, they are
“capitalists” through their pension funds. Many of them are themselves bosses and have “subordinates

---

Result 7: From `The Effective Executive-2002.pdf`, Page 8

. The physician was the knowledge worker, with the nurse as his aide. In other words, up to recent
times, the major problem of organization was efficiency in the performance of the manual worker who
did what he had been told to do. Knowledge workers were not predominant in organization. In fact,
only a small fraction of the knowledge workers of earlier days were part of an organization. Most
of them worked by themselves as professionals, at best with a clerk

---

Result 8: From `The Daily Drucker-2004.pdf`, Page 367

. They have both mobility and self-confidence. This means they have to be treated and managed as
volunteers, in the same way as volunteers who work for not-for-profit organizations. The first thing
such people want to know is what the company is trying to do and where it is going. Next, they are
interested in personal achievement and personal responsibility—which means they have to be put in
the right job. Knowledge workers expect continuous learning and continuous training

---

Result 9: From `The Daily Drucker-2004.pdf`, Page 290

. Given this competitive struggle, a growing number of highly successful knowledge workers—business
managers, university teachers, museum directors, doctors —“plateau” in their forties. If their work
is all they have, they are in trouble. Knowledge workers therefore need to develop some serious
outside interest.   ACTION POINT: Develop a serious satisfying outside interest. Managing in the
Next Society

Result 10: From `The Essential Drucker-2008.pdf`, Page 207

```
. The newly emerging dominant group are “knowledge workers.” Knowledge wor
kers  amount to a third or
more of the workforce in the United States, that is, to as large a proport
ion as  industrial blue-
collar workers ever were, except in wartime. The majority of knowledge wor
kers are  paid at least as
well as blue-collar workers ever were, or better. And the new jobs offer m
uch greater  opportunities
to the individual
──────────────────────────────────────────────────────────────────────────
──────
```

Result 11: From `The Daily Drucker-2004.pdf`, Page 796

```
.’ He is thoroughly at home in economics, political science, industrial ps
ychology, and industrial
sociology, and has succeeded admirably in harmonizing the findings of all
four disciplines and
applying them meaningfully to the practical problems of the ‘enterprise.’
” Drucker believes that
the interests of the worker, management, and corporation are reconcilable
with society. He advances
the idea of “the plant community” in which workers are encouraged to take
on more responsibility and
act like “managers
──────────────────────────────────────────────────────────────────────────
──────
```

6.5 Comparing Embedding Models in Retrieval (optional)

Different embedding models can be used to encode text for retrieval, each offering unique trade-offs in terms of semantic richness, dimensionality, and computational efficiency. In this case, we compare:

- `text-embedding-3-large` for the Book_data collection
- `sentence-transformers/all-MiniLM-L6-v2` for the Book_data_HF_MiniLM collection

Two models vary in architecture, embedding size, and performance characteristics, impacting both retrieval accuracy and system efficiency.

- `text-embedding-3-large` embedding size is 3072, capturing more nuanced semantic details but with higher storage and computational cost.
- `all-MiniLM-L6-v2` embedding size is 384, offering a lightweight alternative with faster processing but may lose some semantic depth in representation.

In [18]:

```
# Run the following command to install langchain-huggingface
# !pip install langchain-huggingface
```

In [19]:

```
from langchain_huggingface import HuggingFaceEmbeddings

# This setup uses the Hugging Face `sentence-transformers` model for text
book_data_vector_store_HF = PGVector(
    embeddings = HuggingFaceEmbeddings(model_name="sentence-transformers/
    # Embedding model: "all-MiniLM-L6-v2" is a lightweight transformer-ba
    collection_name = "Book_data_HF_MiniLM",
    # Name of the collection (table) where embeddings will be stored in t

    connection=shared_connection_string,
    use_jsonb=True,
)

print("PGVector store initialized successfully with Hugging Face embeddin
```

```
PGVector store initialized successfully with Hugging Face embeddings!
```

In [26]:

```
# Perform similarity search using two different embedding models
# Compare retrieval results from both vector stores

# Retrieve the top 3 most relevant documents using `text-embedding-3-larg
book_data_results1 = book_data_vector_store.similarity_search_with_score(

# Retrieve the top 3 most relevant documents using Hugging Face's `all-Mi
book_data_results2 = book_data_vector_store_HF.similarity_search_with_sco

# Iterate through both result sets simultaneously
for i, ((doc1, score1), (doc2, score2)) in enumerate(zip(book_data_result
    display(Markdown(f"## Result {i}"))

    for model_name, doc, score in [
        ("text-embedding-3-large", doc1, score1),
        ("all-MiniLM-L6-v2", doc2, score2)
    ]:
        display(Markdown(f"**From `{doc.metadata['source']}`, Page {doc.m
        display(Markdown(f"**(`{model_name}`** , Similarity Distance: {sc
        print(textwrap.fill(doc.page_content, width=100))
        print("-" * 80 if model_name == "text-embedding-3-large" else "="
```

Result 1

From `The Essential Drucker-2008.pdf`, Page 196

(`text-embedding-3-large` , Similarity Distance: 0.3955)

```
. Knowledge  workers, after all, first came into being in any substantial
numbers a generation ago.
(I coined the  term “knowledge worker” years ago.)   But also the shift fr
om manual workers who do
as they are being told—either by the task or by the  boss—to knowledge wor
kers who have to manage
themselves profoundly challenges social  structure
--------------------------------------------------------------------------
------
```

From `The Daily Drucker-2004.pdf`, Page 797

(`all-MiniLM-L6-v2` , Similarity Distance: 0.3674)

```
. The first part of the book treats the philosophical shift from a Cartesian universe of mechanical
cause to a new universe of pattern, purpose, and configuration. Drucker discusses the need to
organize men of knowledge and of high skill for joint effort, and performance as a key component of
this change
====================================================================================================
```

Result 2

From `The Daily Drucker-2004.pdf`, Page 132

(`text-embedding-3-large` , Similarity Distance: 0.3987)

```
. They do not come with a merger or an acquisition. It is certain that the emergence of the
knowledge worker will bring about fundamental changes in the very structure and nature of the
economic system.   ACTION POINT: What percentage of your workforce consists of people whose work
requires advanced schooling? Tell these people you value their contributions and ask them to
participate in decisions where their expertise is important. Make them feel like owners. Management
Challenges for the 21st Century
--------------------------------------------------------------------------------
```

From `The Daily Drucker-2004.pdf`, Page 800

(`all-MiniLM-L6-v2` , Similarity Distance: 0.3946)

```
. Drucker conveys his life story—from
====================================================================================================
```

Result 3

From `The Effective Executive-2002.pdf`, Page 17

(`text-embedding-3-large` , Similarity Distance: 0.4134)

```
.  It takes his knowledge and uses it as the resource, the motivation, and the vision of  other
knowledge workers. Knowledge workers are rarely in phase with each other, precisely because they
are knowledge workers. Each has his own skill and his own  concerns. One man may be interested in
tax accounting or in bacteriology, or in training  and developing tomorrow’s key administrators in
the city government
--------------------------------------------------------------------------------
```

From `The Daily Drucker-2004.pdf`, Page 12

(`all-MiniLM-L6-v2` , Similarity Distance: 0.4040)

. Steve had the idea for The Daily Drucker. Leah Spiro provided detailed advice and support in writing the book. I am especially grateful for the help Leah provided in reviewing each reading and helping to draft the Action Points. Ceci Hunt copyedited the manuscript. I am grateful both for her skill and hard work. I also owe a debt of gratitude to Diane Aronson, copy chief, and to Knox Huston of HarperCollins for their help in preparing this book

==