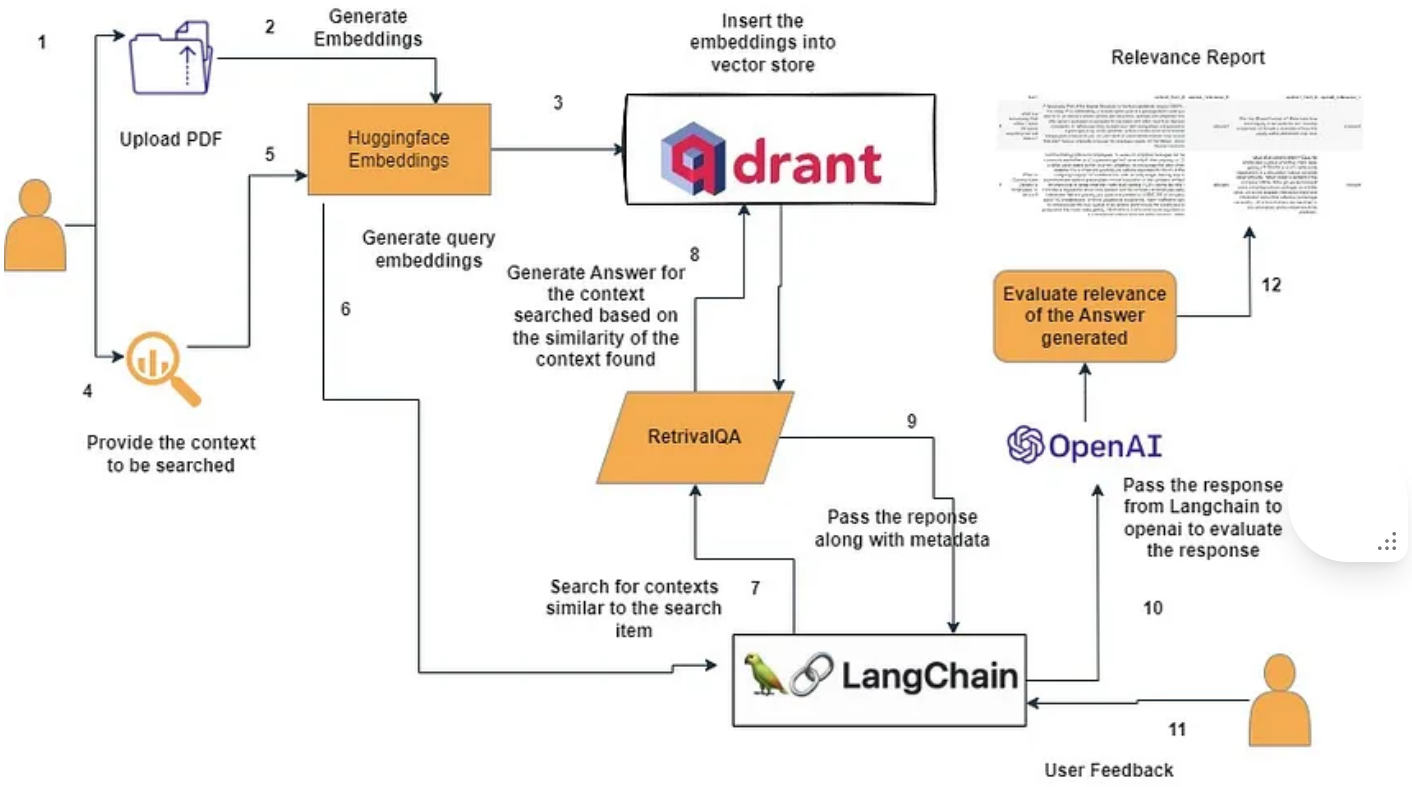
Retrieval Augmented Generation using Qdrant HuggingFace embeddings and Langchain and Evaluate the Response Generated using OpenAI

Plaban Nayak · Follow

Published in AI Planet · 65 min read · Jul 30



What is RAG(Retrieval Augmented Generation) ?

Retrieval Augmented Generation (RAG) is a natural language processing (NLP) technique that combines two fundamental tasks in NLP: information retrieval and text generation. It aims to enhance the generation process by incorporating information from external sources through retrieval. The goal of RAG is to produce more accurate and contextually relevant responses in text generation tasks.

In traditional text generation models like GPT-3, the model generates text based on patterns learned from a large corpus of data, but it may not always have access to specific, up-to-date, or contextually relevant information. Retrieval Augmented Generation addresses this limitation by introducing an information retrieval component.

Here's how RAG works:

Retrieval: The model performs a retrieval step to gather relevant information from external sources. These sources could include a database, a knowledge base, a set of documents, or even search engine results. The retrieval process aims to find snippets or passages of text that contain information related to the given input or prompt.

Augmentation: The retrieved information is then combined with the original input or prompt, enriching the context available to the model for generating the outp By incorporating external knowledge, the model can produce more informed ar accurate responses.

Generation: Finally, the model generates the response, taking into account the retrieved information and the original input. The presence of this additional context helps the model produce more contextually appropriate and relevant outputs.

RAG can be beneficial in various NLP tasks, such as question-answering, dialogue generation, summarization, and more. By incorporating external knowledge, RAG models have the potential to provide more accurate and informative responses compared to traditional generation models that rely solely on the data they were trained on.

What are the advantages of using RAG ?

Retrieval Augmented Generation (RAG) offers several advantages over traditional text generation models, especially in scenarios where access to external information is beneficial. Some of the key advantages include:

*1 . **Contextual Relevance**: RAG models can produce responses that are more contextually relevant and informative. By incorporating information from*

external sources, the generated text is better grounded in real-world facts and up-to-date knowledge, leading to more accurate and context-aware responses.

*2. **Fact Checking and Verification**: Since RAG models retrieve information from reliable external sources, they can perform fact-checking and verification during the generation process. This helps in reducing the generation of false or misleading information and ensures the accuracy of the generated content.*

*3. **Improved Knowledge Incorporation**: RAG models can effectively utilize external knowledge bases or documents to enhance their responses. This is particularly useful in question-answering tasks, where the model can access relevant information from a wide range of sources to provide well-informed and accurate answers.*

*4. **Flexibility and Adaptability**: The ability to retrieve information from diverse sources makes RAG models more flexible and adaptable. They can handle a wide range of topics and tasks without requiring explicit fine-tuning for each specific scenario, as long as the retrieval mechanism is designed to access the relevant information.*

*5. **Handling Out-of-Distribution Inputs**: Traditional text generation models may struggle when faced with out-of-distribution or uncommon inputs that were not present in their training data. RAG models, on the other hand, can leverage the retrieval component to find relevant information, even for unseen or less common inputs.*

*6. **Controlled Content Generation**: RAG models can also be used for content-controlled generation. By guiding the retrieval process and specifying the sources, developers can control the type and quality of information the model uses to generate responses.*

*7. **Reduced Bias**: In some cases, the retrieval mechanism can help reduce bias in generated content. By using diverse sources of information, the model can provide a more balanced and unbiased response, compared to traditional models that may be influenced by the biases present in their training data.*

While RAG offers significant advantages, it's important to be aware of potential challenges and considerations, such as ensuring the reliability of the retrieval sources, handling contradictory information from different sources, and balanc the trade-off between retrieval accuracy and computational efficiency.

Overall, Retrieval Augmented Generation is a promising direction in NLP research, providing a powerful approach to enhance the capabilities of text generation models and making them more useful and trustworthy in real-world applications.

Implementaion using Odrant + Langchain + Huggingface + OpenAI

Here we will implement a Document Search application. Steps involved:-

1. *User Upload a document in .pdf format(either Text Based PDF or Scanned PDF)*
2. *The text from the PDF is retrieved using PyPDF2 package and the text is passed to HuggingFace Embeddings.*
3. *The embeddings along with metadata information like text chunk and page number is inserted into Vector Store in our case we have used Qdrant.*
4. *The user also provides a search query or instruction.*
5. *This query is passed on to the Langchain where it gets converted into query embeddings and then the Vector Store is searched for the contexts matching the query embeddings.*

6. *In our case we search for top 5 best matches based on semantic similarity. This context is then passed on the LLM model which on our case is OpenAI .*
7. *OpenAI takes the matched contexts and generates an answer based on the context supplied.*
8. *The Answer generated is then evaluated for the relevancy using OpenAI.*

Code Implementation :

Install Necessary packages:

```
! pip  install -qU openai langchain PyPDF2 transformers tiktoken pinecone-client
```

Install necessary packages for OCRing

```
!apt install tesseract-ocr
! apt-get install poppler-utils
! pip install -U pytesseract
! pip install pdf2image
```

Upload the required PDF. In my case I have uploaded a document regarding Employee Stock Exchange best practices.

Import Necessary Package

```
from qdrant_client import models, QdrantClient
from langchain.embeddings import HuggingFaceEmbeddings
```

```
from langchain.vectorstores.qdrant import Qdrant
import PyPDF2
from PyPDF2 import PdfReader
from tqdm.auto import tqdm
from uuid import uuid4
from langchain.text_splitter import RecursiveCharacterTextSplitter
from transformers import pipeline
from sentence_transformers import SentenceTransformer, util
import numpy as np
import pandas as pd
from time import time,sleep
import openai
from langchain import VectorDBQA
from langchain.chains import RetrievalQA
from langchain.llms import OpenAI
from langchain.chat_models import ChatOpenAI
from langchain import PromptTemplate
#
import os
import json
#
import io
import pytesseract
import shutil
import random
from PIL import Image
from pdf2image import convert_from_path
from configparser import ConfigParser
#
from typing import Dict, List, Optional, Tuple
from langchain.docstore.document import Document
from langchain.vectorstores import Qdrant
import pandas as pd
import numpy as np
#
```

Set OpenAI keys

```
import openai
from getpass import getpass
import os
os.environ["OPENAI_API_KEY"] =getpass()
```

Define Helper Functions

```
#Checke Whether the document uploaded is a scaned document
def is_scanned_pdf(file_path):
    page_num  = []
    page_content = []
    pdf_reader = PdfReader(file_path)
    text = ""
    for i,page in enumerate(pdf_reader.pages):
      page_num.append(f"Page{i+1}")
      page_content.append(page.extract_text())
      text += page.extract_text()

    # if no text found, it is likely a scanned pdf
    if text == "":
        print("Scanned PDF")
        return True,page_num,page_content

    else:
        print("Text-based PDF")
        return False,page_num,page_content
#
```

```
#Convert Scanned Document to Image
def convert_pdf_to_image(pdf_path):
    pages = convert_from_path(pdf_path, 350)
    #
    output_dir="pdf_img_dir"
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)
    i = 1
    for page in pages:
        image_name = "Page_" + str(i) + ".jpg"
        image_path = os.path.join(output_dir,image_name)
        page.save(image_path, "JPEG")
        i = i+1
    return output_dir
```

```
# Extract text from image
#
```

```
def extract_text_from_image(image_path):
    extractedInformation = pytesseract.image_to_string(Image.open(image_path))
    return extractedInformation
#
```

```
# Compute Cosine Similarity
def get_scores(sentence1,sentence2,smodel):
  # encode sentences to get their embeddings
  embedding1 = smodel.encode(sentence1, convert_to_tensor=True)
  embedding2 = smodel.encode(sentence2, convert_to_tensor=True)
  # compute similarity scores of two embeddings
  cosine_scores = util.pytorch_cos_sim(embedding1, embedding2)
  return cosine_scores.numpy().tolist()[0][0]
#
```

```
#Text Chuncking
def get_text_chunks(text):
    text_splitter = RecursiveCharacterTextSplitter(
                            chunk_size=1000,
                            chunk_overlap=200,
                            separators=["\n\n", "\n", " ", ""]
                            )
    chunks = text_splitter.split_text(text)
    return chunks
#
```

```
# Data preprocessing step
def preprocess_function(request_data,content_type=None):
  pdf_docs = request_data["pdf1"]
  prompt = request_data["query"]
  flag,page_num,page_content = is_scanned_pdf(pdf_docs)
  pdf_name = "temppdf"
  f_path = os.path.join("/tmp", pdf_name)
  #
  print(pdf_docs)
  if flag:
    with open(f_path, "wb") as f:
        f.write(pdf_docs.getbuffer())
    print(f"filesize : {os.path.getsize(f_path)}")
    page_num  = []
```

```
    page_content = []
    text = ""
    # convert pdf to image
    image_dir = convert_pdf_to_image(f_path)
    for i,file in enumerate(os.listdir(image_dir)):
      page_num.append(f"Page{i+1}")
      image_path = os.path.join(image_dir,file)
      text += extract_text_from_image(image_path)
      page_content.append(extract_text_from_image(image_path))

      print(f"Text read from image: {text}")
  #
  metadatas = []
  texts = []
  for p,record in tqdm(zip(page_num,page_content)):
        # first get metadata fields for this record
        metadata = {'page number': p}
        # now we create chunks from the record text
        record_texts = get_text_chunks(record)
        # create individual metadata dicts for each chunk
        record_metadatas = [{
            "chunk": j, "text": text, **metadata
        } for j, text in enumerate(record_texts)]
        # append these to current batches
        texts.extend(record_texts)
        metadatas.extend(record_metadatas)
  c = []
  for i in range(len(metadatas)):
    c.append(metadatas[i]['text'])

  page_content = c
  return (flag,page_num,page_content,metadatas,prompt)
#
```

```
# Load the required embeddings
def model_load_function(model_path):
  embeddings = HuggingFaceEmbeddings(model_kwargs = {'device': 'cpu'},
                                     encode_kwargs = {'normalize_embeddings': Fal
  smodel = SentenceTransformer('sentence-transformers/all-mpnet-base-v2')
  #
  return (embeddings,model_path,smodel)
#
```

```
# The Prediction function , it returns the response as well as the embeddinbgs u
def predict_function(context,model,content_type=None):
  embeddings,model_path,smodel= model
  flag,page_num,page_content,metadatas,prompt = context
  #create vector store
  #

  doc_store = Qdrant.from_texts(page_content,
                                metadatas=metadatas,
                                embedding=embeddings,
                                location=":memory:",
                                prefer_grpc=True,

                                collection="doc_search")

  #query vector store
  prompt_template = """Use the following pieces of context to answer the questio
  Please provide an answer which is factually correct and based on the informati
  Please also mention any quotes supporting the answer if any present in the con

{context}

QUESTION:```{question}```
ANSWER:
  """
  PROMPT = PromptTemplate(
    template=prompt_template, input_variables=["context","question"]
  )
  #
  chain_type_kwargs = {"prompt": PROMPT}
  #
  qa = RetrievalQA.from_chain_type(llm=ChatOpenAI(model_name='gpt-3.5-turbo-16k'
                                                  openai_api_key=os.environ["OPE
                                                  temperature=0),
                                   chain_type="stuff",
                                   chain_type_kwargs={"prompt": PROMPT},
                                   retriever=doc_store.as_retriever(search_kwargs
                                       return_source_documents=True
                                   )



  # Serach for answer
  print(prompt)
  final_result = []
  for pmpt in prompt:
    result = qa(pmpt)
    final_result.append(result)
```

```
  return (final_result,smodel,context,embeddings)
#
```

```
#Processing the Response
#
def postprocess_function(res, content_type=None):
  results,smodel,context,embeddings = res
  flag,page_num,page_content,metadatas,prompt = context
  #
  queries =[]
  answers = []
  final_ref = []
  final_pages = []
  final_scores = []
  for result in results :
    query = result['query']
    answer = result['result']
    queries.append(query)
    answers.append(answer)
    referrences = []
    pages = []
    scores = []
    #
    for i in range(len(result["source_documents"])):
      print(f"Content : {result['source_documents'][i].page_content}")
      referrences.append(result['source_documents'][i].page_content)
      print(f"Page Number : {result['source_documents'][i].metadata['page number
      pages.append(result['source_documents'][i].metadata['page number'])

    final_ref.append(referrences)
    final_pages.append(pages)
    # Get the similarity scores for the documents selected

    for item in referrences:
      scores.append(get_scores(query,item,smodel))
    final_scores.append(scores)
  # Format the response
  predictions = {"Query":queries,
                 "Answer":answers,
                 "Referrences":final_ref,
                 "Page Numbers":final_pages,
                 "Scores": final_scores}
  #Generate Database dataframe
  t = []
  e = []
  for ts in page_content:
```

```
    t.append(ts)
    e.append(embeddings.embed_documents(ts)[0])
  database_df = pd.DataFrame({"text":t,"text_vector":e})
  #database_df.to_parquet("database_df.parquet",index=False)
  #processing quer_df attributes
  txt = []
  text_vector = []
  response = []
  context_text_0 = []
  context_text_1 = []
  context_similarity_0 = []
  context_similarity_1 = []
  user_feedback = []

  for q,a,refs,scrs in zip(queries,answers,final_ref,final_scores):
    txt.append(q)
    text_vector.append(embeddings.embed_query(q))
    response.append(a)
    context_text_0.append(refs[0])
    context_text_1.append(refs[1])
    context_similarity_0.append(scrs[0])
    context_similarity_1.append(scrs[1])
    user_feedback.append(1.0)
  #Create query dataframe based on the response received
  query_df = pd.DataFrame({"text":txt,
                           "text_vector":text_vector,
                           "response":response,
                           "context_text_0":context_text_0,
                           "context_text_1":context_text_1,
                           "context_similarity_0":context_similarity_0,
                           "context_similarity_1":context_similarity_1,
                           "user_feedback":user_feedback})
  # delete temp directories if any created
  output_dir="pdf_img_dir"
  if os.path.exists(output_dir):
    for  file in os.listdir(output_dir):
      file_name = os.path.join(output_dir,file)
      if os.path.exists(file_name) and os.path.isfile(file_name):
        print(f"File {file_name} exists. Hence Removing")
        os.remove(file_name)
  return (predictions,database_df,query_df)
```

Data Preprocessing

```
%%time
# We have passes a set of queries corresponding to the pdf uploaded
request_data = {"pdf1":"/content/Employee-Stock-Option-Plans-ESOP-Best-Practices
                "query":["what is a neccessary Part of the Capital Structure acc
                         "When to Communicate Options to Employees: % versus $",
                         "When to Create an ESOP?",
                         "What are the Common Terms in an Options Package?"]}
content = preprocess_function(request_data,content_type=None)
```

```
Text-based PDF
/content/Employee-Stock-Option-Plans-ESOP-Best-Practices-2.pdf
44/? [00:00<00:00, 1948.29it/s]
CPU times: user 1.49 s, sys: 7.88 ms, total: 1.5 s
Wall time: 1.52 s
```

Load the Model

```
%%time
model = model_load_function(model_path=None)
```

```
CPU times: user 2.3 s, sys: 1.93 s, total: 4.22 s
Wall time: 8.65 s
```

Invoke Predictions

```
%%time
response = predict_function(content,model,content_type=None)
```

```
#response
([{'query': 'what is a neccessary Part of the Capital Structure according to Fre
   'result': 'According to Fred Wilson, a necessary part of the capital structur
   'source_documents': [Document(page_content="A Necessary Part of the Capital S
    Document(page_content='The Top -Down Process  \n1.Determine how much equity
    Document(page_content='Resources & Further Reading  \n•Fred Wilson / Union S
    Document(page_content='Resources & Further Reading  \n•Andy Rachleff  / Weal
    Document(page_content='Late-VC Flush with capital,  startups at this stage \
  {'query': 'When to Communicate Options to Employees: % versus $',
   'result': 'The context does not provide information about when to communicate
   'source_documents': [Document(page_content='Communicating Options to Emplo
    Document(page_content='value of an options grant  \n•Easy for employees
    Document(page_content='Why Issue Options to Employees?  \n•Attract Talen
    Document(page_content='0.8 An Example: Hiring a CTO  \nEmployee Information
    Document(page_content='3. Determine the Dollar Value of the Options Grant  \
  {'query': 'When to Create an ESOP?',
   'result': 'ESOPs should be created during the seed stage or early-VC stage of
   'source_documents': [Document(page_content='When to Create an ESOP?  \nStage
    Document(page_content='Two Approaches  \nIn reality, creating an ESOP will r
    Document(page_content='members  •Founders  are not top \nmanagers in the bus
    Document(page_content='Legal Advice  \nWhen structuring an ESOP, engage an e
    Document(page_content='ESOPS FOR THE LONG TERM  Part IV', metadata={'chunk':
  {'query': 'What are the Common Terms in an Options Package?',
   'result': 'The common terms in an options package are:\n1. Number of Shares:
   'source_documents': [Document(page_content='Common Terms in an Options Packag
    Document(page_content='Appropriate use of these securities will \nvary based
    Document(page_content='What is an Option?  \n•Why do options have intrinsic
    Document(page_content='value of an options grant  \n•Easy for employees to g
    Document(page_content='Strike Price  \n•Options are “struck” at a specific s
 SentenceTransformer(
   (0): Transformer({'max_seq_length': 384, 'do_lower_case': False}) with Transf
   (1): Pooling({'word_embedding_dimension': 768, 'pooling_mode_cls_token': Fals
   (2): Normalize()
 ),
 (False,
  ['Page1',
   'Page2',
   'Page3',
   'Page4',
   'Page5',
   'Page6',
   'Page7',
   'Page8',
   'Page9',
   'Page10',
   'Page11',
   'Page12',
   'Page13',
```

```
 'Page14',
 'Page15',
 'Page16',
 'Page17',
 'Page18',
 'Page19',
 'Page20',
 'Page21',
 'Page22',
 'Page23',
 'Page24',
 'Page25',
 'Page26',
 'Page27',
 'Page28',
 'Page29',
 'Page30',
 'Page31',
 'Page32',
 'Page33',
 'Page34',
 'Page35',
 'Page36',
 'Page37',
 'Page38',
 'Page39',
 'Page40',
 'Page41',
 'Page42',
 'Page43',
 'Page44'],
['Startup Employee Stock \nOptions Plans (ESOPs)  \n \nOverview and Best Pract
 'Table of Contents  \nPart I: Intro to Options Plans  \n•What is an ESOP?  \n
 '•Important Takeaways  Part III : The Fine Print - Terms  \n•Strike Price  \n
 'INTRO TO OPTIONS PLANS  Part I',
 'What is an ESOP?  \n•An Employee Stock Options Plan (ESOP)  \n \n•An allocat
 'What is an Option?  \n•Why do options have intrinsic value?  \n-A effective
 'Appropriate use of these securities will \nvary based on local regulatory an
 'Lifecycle of a Startup ESOP  \n•Founders and early investors  create an ESOP
 'Common Terms in an Options Package  \nNumber of Shares  The total number of
 'Why Issue Options to Employees?  \n•Attract Talent:  options can be used to
 'The defining difference between Silicon Valley companies and almost \nevery
 "A Necessary Part of the Capital Structure  \n•Venture capitalists require ES
 'When to Create an ESOP?  \nStage  Considerations  Takeaway  \nPre-seed  Foun
 'Late-VC Flush with capital,  startups at this stage \nbegin to steadily ramp
 'Communicating Options to Employees: % versus $  \nOptions packages can be co
 'value of an options grant  \n•Easy for employees to grasp  what they \nare r
 'HOW MUCH TO GRANT  Part II',
 'Two Approaches  \nIn reality, creating an ESOP will require a combination of
 'The Top -Down Process  \n1.Determine how much equity to set aside for non -f
```

```
 '1. How Much Equity to Set Aside in the ESOP?  \n•Founders  are top managers
 'members  •Founders  are not top \nmanagers in the business, or \nplan to ste
 '2. A Typical Distribution Schedule  \nSeniority  Equity Allocation  \nFirst
 'The Bottom -Up Process  \n1.Segment your human resources  \n2.Create pay mul
 '1. Segment Your Human Resources  \nSegment  Roles  Award  \nin % or $?  \nFo
 '2. Establish Pay Multipliers for Each Role  \nSegment  Roles  Award  \nin %
 '3. Determine the Dollar Value of the Options Grant  \nEmployee  \nBase Salar
 '4. Determine the Current Share Price  \nLatest “True” Valuation  \nFully -Di
 '5. Calculate the Options Grant  \nOptions Grant  \n(Number of Shares)  = Dol
 '0.8 An Example: Hiring a CTO  \nEmployee Information  \n•CTO (Senior Team)
 'Important Takeaways  \n•Top-down planning (“the allocation”)  \n-Holisticall
 'THE FINE PRINT - TERMS  Part III',
 'Strike Price  \n•Options are “struck” at a specific strike price when is
 'Vesting Schedule  \n•Vesting protects the company  \n-It stages the acc
 'The Cliff  \n•The cliff protects the company  \n•This trial period exists
 '200 shares400 shares600 shares800 shares1,000 shares\n0 months 12 months 24
 'Vesting in a Liquidity Event  \nThe terms of an options package should speci
 'Exercising Options  \n•Options are “in the money” if the share value is abov
 'Tax Considerations  \nOptions can have material tax consequences for employe
 'Legal Advice  \nWhen structuring an ESOP, engage an experienced \nstartup la
 'ESOPS FOR THE LONG TERM  Part IV',
 'Retention Grants  \n•Retention grants are used to incentive employees to sta
 'Discretionary Grants  \n•Bonuses Outstanding Performance  \n-These discretio
 'Social Impact Considerations  \n•Companies focused on social impact goals an
 'Options Modeling - Overview  \n•Creating an ESOP has lasting implications on
 'Options Modeling - A Detailed Example  \nEmployee Stock Options Plan (ESOP)
 'Year 2 Hires\nSales Director $70,000 0.30x $21,000 42 0.42% 42 0.35%\nSalesp
 'RESOURCES & FURTHER \nREADING  Part V',
 'Resources & Further Reading  \n•Fred Wilson / Union Square Ventures Series o
 'Resources & Further Reading  \n•Andy Rachleff  / Wealthfront  materials  \n-
 'This work  is licensed  under  the Creative  Commons  Attribution -\nShareAl
[{'chunk': 0,
  'text': 'Startup Employee Stock \nOptions Plans (ESOPs)  \n \nOverview and B
  'page number': 'Page1'},
 {'chunk': 0,
  'text': 'Table of Contents  \nPart I: Intro to Options Plans  \n•What is an
  'page number': 'Page2'},
 {'chunk': 1,
  'text': '•Important Takeaways  Part III : The Fine Print - Terms  \n•Strike
  'page number': 'Page2'},
 {'chunk': 0,
  'text': 'INTRO TO OPTIONS PLANS  Part I',
  'page number': 'Page3'},
 {'chunk': 0,
  'text': 'What is an ESOP?  \n•An Employee Stock Options Plan (ESOP)  \n \n•A
  'page number': 'Page4'},
 {'chunk': 0,
  'text': 'What is an Option?  \n•Why do options have intrinsic value?  \n-A e
  'page number': 'Page5'},
 {'chunk': 1,
```

```
 'text': 'Appropriate use of these securities will \nvary based on local regu
 'page number': 'Page5'},
{'chunk': 0,
 'text': 'Lifecycle of a Startup ESOP  \n•Founders and early investors  creat
 'page number': 'Page6'},
{'chunk': 0,
 'text': 'Common Terms in an Options Package  \nNumber of Shares  The total n
 'page number': 'Page7'},
{'chunk': 0,
 'text': 'Why Issue Options to Employees?  \n•Attract Talent:  options can be
 'page number': 'Page8'},
{'chunk': 0,
 'text': 'The defining difference between Silicon Valley companies and al
 'page number': 'Page9'},
{'chunk': 0,
 'text': "A Necessary Part of the Capital Structure  \n•Venture capitalists r
 'page number': 'Page10'},
{'chunk': 0,
 'text': 'When to Create an ESOP?  \nStage  Considerations  Takeaway  \nPre-s
 'page number': 'Page11'},
{'chunk': 1,
 'text': 'Late-VC Flush with capital,  startups at this stage \nbegin to stea
 'page number': 'Page11'},
{'chunk': 0,
 'text': 'Communicating Options to Employees: % versus $  \nOptions packages
 'page number': 'Page12'},
{'chunk': 1,
 'text': 'value of an options grant  \n•Easy for employees to grasp  what the
 'page number': 'Page12'},
{'chunk': 0, 'text': 'HOW MUCH TO GRANT  Part II', 'page number': 'Page13'},
{'chunk': 0,
 'text': 'Two Approaches  \nIn reality, creating an ESOP will require a combi
 'page number': 'Page14'},
{'chunk': 0,
 'text': 'The Top -Down Process  \n1.Determine how much equity to set aside f
 'page number': 'Page15'},
{'chunk': 0,
 'text': '1. How Much Equity to Set Aside in the ESOP?  \n•Founders  are top
 'page number': 'Page16'},
{'chunk': 1,
 'text': 'members  •Founders  are not top \nmanagers in the business, or \npl
 'page number': 'Page16'},
{'chunk': 0,
 'text': '2. A Typical Distribution Schedule  \nSeniority  Equity Allocation
 'page number': 'Page17'},
{'chunk': 0,
 'text': 'The Bottom -Up Process  \n1.Segment your human resources  \n2.Creat
 'page number': 'Page18'},
{'chunk': 0,
 'text': '1. Segment Your Human Resources  \nSegment  Roles  Award  \nin % or
```

```
 'page number': 'Page19'},
{'chunk': 0,
 'text': '2. Establish Pay Multipliers for Each Role  \nSegment  Roles  Award
 'page number': 'Page20'},
{'chunk': 0,
 'text': '3. Determine the Dollar Value of the Options Grant  \nEmployee  \nB
 'page number': 'Page21'},
{'chunk': 0,
 'text': '4. Determine the Current Share Price  \nLatest “True” Valuation  \n
 'page number': 'Page22'},
{'chunk': 0,
 'text': '5. Calculate the Options Grant  \nOptions Grant  \n(Number of Share
 'page number': 'Page23'},
{'chunk': 0,
 'text': '0.8 An Example: Hiring a CTO  \nEmployee Information  \n•CTO (Se.
 'page number': 'Page24'},
{'chunk': 0,
 'text': 'Important Takeaways  \n•Top-down planning (“the allocation”)  \n-Ho
 'page number': 'Page25'},
{'chunk': 0,
 'text': 'THE FINE PRINT - TERMS  Part III',
 'page number': 'Page26'},
{'chunk': 0,
 'text': 'Strike Price  \n•Options are “struck” at a specific strike price wh
 'page number': 'Page27'},
{'chunk': 0,
 'text': 'Vesting Schedule  \n•Vesting protects the company  \n-It stages the
 'page number': 'Page28'},
{'chunk': 0,
 'text': 'The Cliff  \n•The cliff protects the company  \n•This trial period
 'page number': 'Page29'},
{'chunk': 0,
 'text': '200 shares400 shares600 shares800 shares1,000 shares\n0 months 12 m
 'page number': 'Page30'},
{'chunk': 0,
 'text': 'Vesting in a Liquidity Event  \nThe terms of an options package sho
 'page number': 'Page31'},
{'chunk': 0,
 'text': 'Exercising Options  \n•Options are “in the money” if the share valu
 'page number': 'Page32'},
{'chunk': 0,
 'text': 'Tax Considerations  \nOptions can have material tax consequences fo
 'page number': 'Page33'},
{'chunk': 0,
 'text': 'Legal Advice  \nWhen structuring an ESOP, engage an experienced \ns
 'page number': 'Page34'},
{'chunk': 0,
 'text': 'ESOPS FOR THE LONG TERM  Part IV',
 'page number': 'Page35'},
{'chunk': 0,
```

```
    'text': 'Retention Grants  \n•Retention grants are used to incentive employe
    'page number': 'Page36'},
   {'chunk': 0,
    'text': 'Discretionary Grants  \n•Bonuses Outstanding Performance  \n-These
    'page number': 'Page37'},
   {'chunk': 0,
    'text': 'Social Impact Considerations  \n•Companies focused on social impact
    'page number': 'Page38'},
   {'chunk': 0,
    'text': 'Options Modeling - Overview  \n•Creating an ESOP has lasting implic
    'page number': 'Page39'},
   {'chunk': 0,
    'text': 'Options Modeling - A Detailed Example  \nEmployee Stock Options
    'page number': 'Page40'},
   {'chunk': 1,
    'text': 'Year 2 Hires\nSales Director $70,000 0.30x $21,000 42 0.42% 42 0.35
    'page number': 'Page40'},
   {'chunk': 0,
    'text': 'RESOURCES & FURTHER \nREADING  Part V',
    'page number': 'Page41'},
   {'chunk': 0,
    'text': 'Resources & Further Reading  \n•Fred Wilson / Union Square Ventures
    'page number': 'Page42'},
   {'chunk': 0,
    'text': 'Resources & Further Reading  \n•Andy Rachleff  / Wealthfront  mater
    'page number': 'Page43'},
   {'chunk': 0,
    'text': 'This work  is licensed  under  the Creative  Commons  Attribution -
    'page number': 'Page44'}],
  ['what is a neccessary Part of the Capital Structure according to Fred Wilson?
   'When to Communicate Options to Employees: % versus $',
   'When to Create an ESOP?',
   'What are the Common Terms in an Options Package?']),
 HuggingFaceEmbeddings(client=SentenceTransformer(
   (0): Transformer({'max_seq_length': 384, 'do_lower_case': False}) with Transf
   (1): Pooling({'word_embedding_dimension': 768, 'pooling_mode_cls_token': Fals
   (2): Normalize()
 ), model_name='sentence-transformers/all-mpnet-base-v2', cache_folder=None, mod
```

Format the response generated

```
%%time
out = postprocess_function(response, content_type=None)
```

```
#
predictions,database_df,query_df = out
print(predictions)
```

```
#response
{'Query': ['what is a neccessary Part of the Capital Structure according to Fred
```

Display Returned Data Frames

```
pd.set_option("display.max_colwidth", 50)
database_df.head()
```

	text	text_vector	centered_text_vector
0	Startup Employee Stock \nOptions Plans (ESOPs)...	[0.0003137296298518777, 0.008552555926144123, ...	[0.0006470922112281248, 0.0088859185075203 7, -...
1	Table of Contents \nPart I: Intro to Options ...	[-0.023618347942829132, -0.0637901052832603 5, ...	[-0.023495769524682058, -0.06366752686511327, ...
2	•Important Takeaways Part III : The Fine Prin...	[-0.018981635570526123, 0.03906095772981644, -...	[-0.01897012570371215, 0.0390724675966304 1, -0...
3	INTRO TO OPTIONS PLANS Part I	[0.011805826798081398, 0.048538338392972946, -...	[0.012143419000686384, 0.0488759305955779 3, -0...
4	What is an ESOP? \n•An Employee Stock Options...	[-0.030949262902140617, 0.045493677258491516, ...	[-0.030975572079112945, 0.0454673680815191 9, 0...

```
query_df.head()
```

	text	text_vector	response	context_text_0	context_text_1	context_similarity_0	context_similarity_1	user_feedback
0	what is a neccessary Part of the Capital Struc...	[0.018048636615276337, 0.004823092371225357, -...	According to Fred Wilson, a necessary part of ...	A Necessary Part of the Capital Structure \n•...	The Top -Down Process \n1.Determine how much ...	0.491406	0.456906	1.0
1	When to Communicate Options to Employees: % ve...	[0.021837031468749046, 0.016419939696788788, -...	The context does not provide information about...	Communicating Options to Employees: % versus $...	value of an options grant \n•Easy for employe...	0.754257	0.569296	1.0
2	When to Create an ESOP?	[0.0076230671256780624, -0.02172561176121235, ...	ESOPs should be created during the seed stage ...	When to Create an ESOP? \nStage Consideratio...	Two Approaches \nIn reality, creating an ESOP...	0.738180	0.700007	1.0
3	What are the Common Terms in an	[-0.023321891203522682, -0.1582910716533661, -	The common terms in an options	Common Terms in an Options Package	Appropriate use of these securities will	0.664095	0.576239	

Process the Data for Relevancy of the answers generated Evaluation

```
cen =[]
c_text_vector =[]
for items in  database_df["text_vector"].values:
  cen.append(np.mean(items))
  c_text_vector.append(items - np.mean(items))
#
database_df["centered_text_vector"]  = c_text_vector
database_df.head()
```

	text	text_vector	centered_text_vector
0	Startup Employee Stock \nOptions Plans (ESOPs)...	[0.0003137296298518777, 0.008552555926144123, ...	[0.0006470922112281248, 0.00888591850752037, -...
1	Table of Contents \nPart I: Intro to Options ...	[-0.023618347942829132, -0.0637901052832603 5, ...	[-0.023495769524682058, -0.06366752686511327, ...
2	•Important Takeaways Part III : The Fine Prin...	[-0.018981635570526123, 0.03906095772981644, -...	[-0.01897012570371215, 0.03907246759663041, -0...
3	INTRO TO OPTIONS PLANS Part I	[0.011805826798081398, 0.048538338392972946, -...	[0.012143419000686384, 0.04887593059557793, -0...
4	What is an ESOP? \n•An Employee Stock Options...	[-0.030949262902140617, 0.045493677258491516, ...	[-0.030975572079112945, 0.04546736808151919, 0...

```
qcen =[]
qc_text_vector =[]
for items in  query_df["text_vector"].values:
  qcen.append(np.mean(items))
  qc_text_vector.append(items - np.mean(items))
#
```

```
query_df["centered_text_vector"] = qc_text_vector
query_df.head()
```

	text	text_vector	response	context_text_0	context_text_1	context_similarity_0	context_similarity_1	user_feedback	centered_text_vector
0	what is a neccessary Part of the Capital Struc...	[0.018048636615276337, 0.004823092371225357, -...	According to Fred Wilson, a necessary part of ...	A Necessary Part of the Capital Structure \n•...	The Top -Down Process \n1.Determine how much ...	0.491406	0.456906	1.0	[0.0180658309004624 7, 0.0048402866564114895, -...
1	When to Communicate Options to Employees: % ve...	[0.021837031468749046, 0.016419939696788788, -...	The context does not provide information about...	Communicating Options to Employees: % versus $...	value of an options grant \n•Easy for employe...	0.754257	0.569296	1.0	[0.021674202866316567, 0.0162571110943563 1,
2	When to Create an ESOP?	[0.0076230671256780624, -0.0217256117612123 5, ...	ESOPs should be created during the seed stage ...	When to Create an ESOP? \nStage Consideratio...	Two Approaches \nIn reality, creating an ESOP...	0.738180	0.700007	1.0	[0.007500557095 -0.021848121791605 ...
3	What are the Common Terms in an Options	[-0.023321891203522682, -0.1582910716533661, -...	The common terms in an options package	Common Terms in an Options Package \nNumber o	Appropriate use of these securities will \nvar	0.664095	0.576239	1.0	[-0.02312235172527599, -0.1580915321751194, 8....

Gauge the relevance of the contexts based on the questions asked as evaluated by GPT4

```
import os
import textwrap
from datetime import timedelta
from typing import Dict, List, Optional, Tuple
from tenacity import (
    retry,
    stop_after_attempt,
    wait_random_exponential,
)
from tqdm import tqdm

pd.set_option("display.max_colwidth", None)
#
EVALUATION_SYSTEM_MESSAGE = "You will be given a query and a reference text. You
QUERY_CONTEXT_PROMPT_TEMPLATE = """# Query: {query}

# Reference: {reference}

# Binary: """

openai.api_key = os.environ["OPENAI_API_KEY"]
@retry(wait=wait_random_exponential(min=1, max=60), stop=stop_after_attempt(6))
def evaluate_query_and_retrieved_context(query: str, context: str, model_name: s
```

```
        prompt = QUERY_CONTEXT_PROMPT_TEMPLATE.format(
            query=query,
            reference=context,
        )
        response = openai.ChatCompletion.create(
            messages=[
                {"role": "system", "content": EVALUATION_SYSTEM_MESSAGE},
                {"role": "user", "content": prompt},
            ],
            model=model_name,
        )
        return response["choices"][0]["message"]["content"]


    def evaluate_retrievals(
        retrievals_data: Dict[str, str],
        model_name: str,
    ) -> List[str]:
        responses = []
        for query, retrieved_context in tqdm(retrievals_data.items()):
            response = evaluate_query_and_retrieved_context(query, retrieved_context
            responses.append(response)
        return responses


    def process_binary_responses(
        binary_responses: List[str], binary_to_string_map: Dict[int, str]
    ) -> List[str]:
        """
        Parse binary responses and convert to the desired format
        converts them to the desired format. The binary_to_string_map parameter
        should be a dictionary mapping binary values (0 or 1) to the desired
        string values (e.g. "irrelevant" or "relevant").
        """
        processed_responses = []
        for binary_response in binary_responses:
            try:
                binary_value = int(binary_response.strip())
                processed_response = binary_to_string_map[binary_value]
            except (ValueError, KeyError):
                processed_response = None
            processed_responses.append(processed_response)
        return processed_responses

    #Invoke the functions defined for Evaluation
    sample_query_df = query_df.head(10).copy()
    evaluation_model_name = "gpt-3.5-turbo"  # use GPT-4 if you have access
    for context_index in range(2):
        retrievals_data = {
            row["text"]: row[f"context_text_{context_index}"] for _, row in sample_q
```

```
        }
        raw_responses = evaluate_retrievals(retrievals_data, evaluation_model_name)
        processed_responses = process_binary_responses(raw_responses, {0: "irrelevan
        sample_query_df[f"openai_relevance_{context_index}"] = processed_responses
sample_query_df[
        ["text", "context_text_0", "openai_relevance_0", "context_text_1", "openai_r
].head(10)
```

Output Evaluations Generated

	text	context_text_0	openai_relevance_0	context_text_1	openai_relevance_1
0	what is a neccessary Part of the Capital Structure according to Fred Wilson?	A Necessary Part of the Capital Structure \n•Venture capitalists require ESOPs . For many VCs, establishing a \nstock option pool is a prerequisite to closing a deal \n–In an industry where options are ubiquitous, startups are compelled \nto offer options packages to compete for top talent with other \nventure -backed companies \n–When operating budgets are tight, competitive compensation \npackages may not be possible; options can be used to incentivize \nemployees instead of cash \nI can't think of a term sheet that we have issued that didn't have a \nspecific provision for employee equity. \nFred Wilson, Union Square Ventures	relevant	The Top -Down Process \n1.Determine how much equity to set aside for non -founder employees \n2.Create a schedule of how this equity will be distributed over time	irrelevant
1	When to Communicate Options to Employees: % versus $	Communicating Options to Employees: % versus $ \nOptions packages can be communicated either as (1) a percentage \nof ownership in the company, or (2) a dollar value based on the \ncurrent valuation; we encourage the latter when possible \n% $ \n"We are granting you options equivalent to \n0.5% of the company's equity" \nConsiderations: \n•At an early stage, the only way to \ncommunicate options grants given no true \nvaluation of the company \n•Hard for employees to grasp what they \nare really getting ("0.5% seems too little") \n•Invites a negotiation about what percent \nof the company an employee really \ndeserves "We are granting you options equivalent to \n$200,000 of company stock" \nConsiderations: \n•Once valuation is established, highly \neffective way to communicate the true \nvalue of an options grant \n•Easy for employees to grasp what they \nare really getting ("$200,000 is a lot") \n•Grounds negotiations	relevant	value of an options grant \n•Easy for employees to grasp what they \nare really getting ("$200,000 is a lot") \n•Grounds negotiations in a discussion \nabout concrete dollar amounts, rather \nthan a percent of the company \nNote: Although we recommend communicating options packages as a dollar value, we do not suggest \nrefusing employees information about their effective percentage ownership ; all options holders are \nentitled to this information, and to refuse would be unethical	relevant
2	When to Create an ESOP?	When to Create an ESOP? \nStage Considerations Takeaway \nPre-seed Founders focused on traction (often too \nbusy for an ESOP). Key employees are \ngiven equity/options on an ad hoc basis ESOP not necessary, but it can be \nhelpful to sanity check how much equity \nyou are giving away to early hires \nSeed First outside financing round. Investors \neither angel or institutional; institutional \ninvestors will require an ESOP Seed rounds can be closed without an \nESOP; the benefit to doing so is that \nseed investors then share in the dilution \nEarly -VC The first true VC round. Investors will \nrequire an ESOP in place. New hires \nwill be seek large equity grants. ESOP must be created (to appease \ninvestors and to serve as a guideline for \nthe size of new -hire options grants) \nLate-VC Flush with capital, startups at this stage \nbegin to steadily ramp -up hiring, yet \nemployees still want equity Important to have standardized the \nESOP and the amount of equity granted	relevant	Two Approaches \nIn reality, creating an ESOP will require a combination of \ntop-down and bottom -up planning Top Down Bottom Up \nDecide the total amount of \nequity to be granted; allocate \nthese shares to employees \nover time Decide the appropriate size of \nindividual equity grants by \nposition; issue these shares as \nemployees are hired	relevant
3	What are the Common Terms in an Options Package?	Common Terms in an Options Package \nNumber of Shares The total number of options granted to an \nemployee, and therefore the maximum number of \nshares that employee has access to \nStrike Price The price the employee must pay to purchase \neach share if and when the employee chooses to \nexercise the option \nVesting Schedule The timeline over which the options become \nwholly owned and exercisable by the employee \n(no longer subject to repurchase by the company) \nCliff Period The trial period during which no vesting occurs; in \nthis period vesting accrues, but the total effect of \nthis vesting is realized immediately after the cliff \nExpiration Date The last date on which the options may be \nexercised and converted into common shares by \nthe employee \nWe will discuss the mechanics in further detail, but these basic	relevant	Appropriate use of these securities will \nvary based on local regulatory and tax \nconsiderations. An option is a right (but not an obligation) to purchase a quantity \nof a company's stock at a set price for a certain period of time	irrelevant

Conclusion:

Here we have demonstrated how to use RAG concept to make the answer generated more factful and evaluate the answer generated for their

relevancy .

References:

QA using a Retriever | 🦜 🔗 Langchain

This example showcases question answering over an index.

python.langchain.com

Semantic Search for Beginners - Qdrant

Qdrant is an Open-Source Vector Database and Vector Search Engine written in Rust. It provides fast and scalable vector...

qdrant.tech

Semantic Similarity Using Transformers

Compute Semantic Textual Similarity between two texts using Pytorch and SentenceTransformers

towardsdatascience.com

Connect with me

Langchain Qdrant OpenAI Vector Database Vector Search

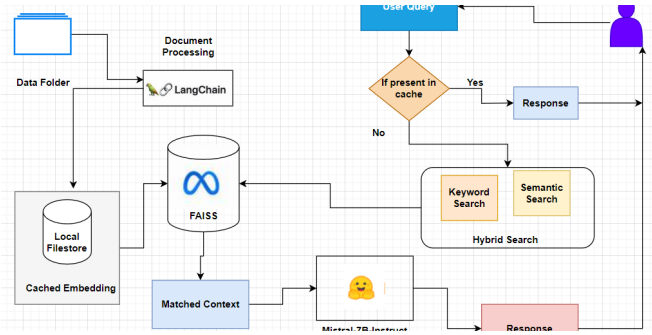
Written by Plaban Nayak

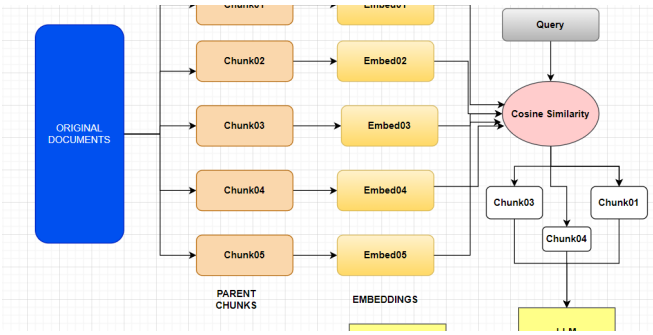
530 Followers · Editor for AI Planet

Machine Learning and Deep Learning enthusiast

Follow

More from Plaban Nayak and AI Planet





Plaban Nayak in AI Planet

Advanced RAG Implementation on Custom Data Using Hybrid Searc...

What is RAG ?

40 min read · Oct 8

350 6

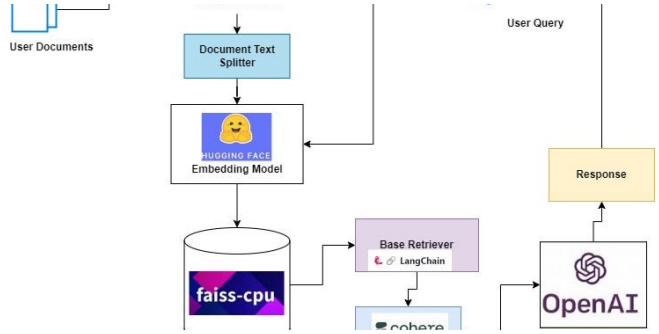
Plaban Nayak in AI Planet

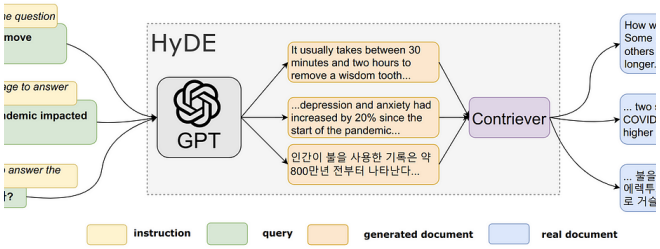
Advanced RAG- Providing Broader Context to LLMs Using...

Traditional RAG Paradigm

10 min read · Oct 29

148





ɪtration of the HyDE model. Documents snippets are shown. HyDE serves all types o
erlying GPT-3 and Contriever/mContriever models.

Plaban Nayak in AI Planet

Advanced RAG- Cohere Re-Ranker

LLMs can acquire new information in at least two ways:

13 min read · Oct 24

38 4

Plaban Nayak in AI Planet

Advanced RAG — Improving retrieval using Hypothetical...

What is HyDE ?

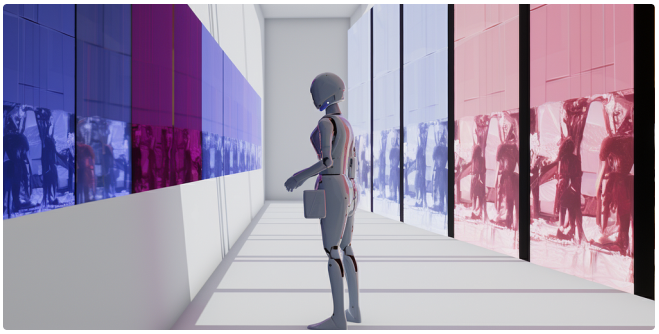
140 min read · Nov 4

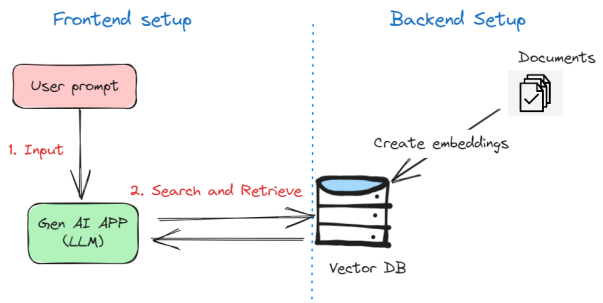
158

See all from Plaban Nayak

See all from AI Planet

Recommended from Medium





Ozgur Guler

Why do RAG pipelines fail? Advanced RAG Patterns—Part1

The failures in RAG pipelines can be attributed to a cascade of challenges...

6 min read · Oct 16

135

NandaKishore Joshi

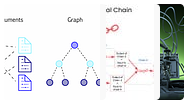
When Retrieval Augmented Generation (RAG) Fails—...

In this article we try to explain the intricacies of RAG as a technique and there by...

4 min read · Oct 23

134 1

Lists



Natural Language Processing

891 stories · 416 saves

AI Regulation

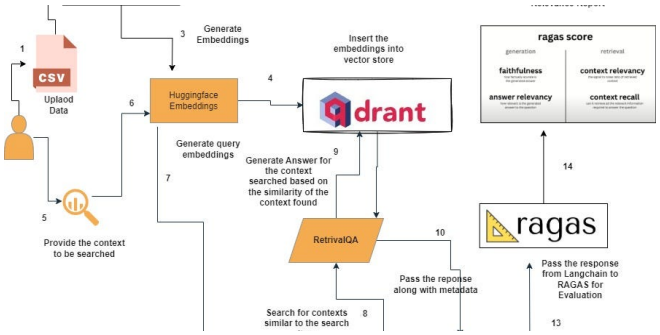
6 stories · 198 saves

Coding & Development

11 stories · 284 saves

Generative AI Recommended Reading

52 stories · 443 saves



Plaban Nayak in AI Planet

Evaluate RAG Pipeline using RAGAS

RAG(Retrieval Augmented Generation)

17 min read · Sep 3

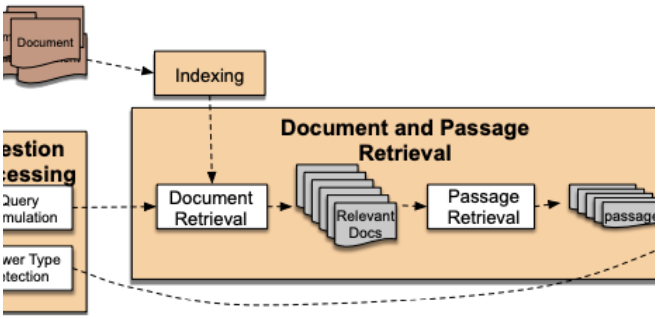
Amogh Agastya in Better Programming

Harnessing Retrieval Augmented Generation With Langchain

Implementing RAG using Langchain

19 min read · Sep 21

115 2



Krishna Yogi

Building a question-answering system using LLM on your private...

11 min read · Oct 5

182 6

388 6

Utkarsh Parashar in GoPenAI

Different types of indexes in LlamaIndex to improve your RAG...

Here we will discuss the different types of indexes LlamaIndex offers to better organize...

11 min read · Oct 6

9

See more recommendations