

1. Install and Import the Required Libraries

```
# Install all the required libraries
```

```
!pip install pdfplumber tiktoken openai chromaDB sentence-transformers -q
```

```

_____ 56.4/56.4 kB 2.0 MB/s eta 0:00:00
_____ 1.8/1.8 MB 35.3 MB/s eta 0:00:00
_____ 262.9/262.9 kB 25.7 MB/s eta 0:00:00
_____ 525.5/525.5 kB 42.4 MB/s eta 0:00:00
_____ 163.3/163.3 kB 17.6 MB/s eta 0:00:00
_____ 5.6/5.6 MB 81.3 MB/s eta 0:00:00
_____ 2.8/2.8 MB 72.7 MB/s eta 0:00:00
_____ 75.6/75.6 kB 6.6 MB/s eta 0:00:00
_____ 2.4/2.4 MB 80.1 MB/s eta 0:00:00
_____ 92.1/92.1 kB 11.5 MB/s eta 0:00:00
_____ 60.8/60.8 kB 7.5 MB/s eta 0:00:00
_____ 41.3/41.3 kB 4.4 MB/s eta 0:00:00
_____ 5.4/5.4 MB 92.1 MB/s eta 0:00:00
_____ 6.8/6.8 MB 81.1 MB/s eta 0:00:00
_____ 60.1/60.1 kB 5.2 MB/s eta 0:00:00
_____ 106.1/106.1 kB 12.1 MB/s eta 0:00:00
_____ 67.3/67.3 kB 7.8 MB/s eta 0:00:00

```

```
Installing build dependencies ... done
```

```
Getting requirements to build wheel ... done
```

```
Preparing metadata (pyproject.toml) ... done
```

```

_____ 698.9/698.9 kB 54.3 MB/s eta 0:00:00
_____ 1.6/1.6 MB 79.4 MB/s eta 0:00:00
_____ 67.6/67.6 kB 8.0 MB/s eta 0:00:00
_____ 144.8/144.8 kB 17.6 MB/s eta 0:00:00
_____ 71.5/71.5 kB 8.4 MB/s eta 0:00:00
_____ 77.9/77.9 kB 9.5 MB/s eta 0:00:00
_____ 58.3/58.3 kB 7.1 MB/s eta 0:00:00
_____ 46.0/46.0 kB 5.2 MB/s eta 0:00:00
_____ 50.8/50.8 kB 6.0 MB/s eta 0:00:00
_____ 23.7/23.7 MB 58.1 MB/s eta 0:00:00
_____ 823.6/823.6 kB 63.6 MB/s eta 0:00:00
_____ 14.1/14.1 MB 84.2 MB/s eta 0:00:00
_____ 731.7/731.7 MB 2.1 MB/s eta 0:00:00
_____ 410.6/410.6 MB 1.3 MB/s eta 0:00:00
_____ 121.6/121.6 MB 8.3 MB/s eta 0:00:00
_____ 56.5/56.5 MB 9.8 MB/s eta 0:00:00
_____ 124.2/124.2 MB 8.3 MB/s eta 0:00:00
_____ 196.0/196.0 MB 2.3 MB/s eta 0:00:00
_____ 166.0/166.0 MB 6.5 MB/s eta 0:00:00
_____ 99.1/99.1 kB 12.6 MB/s eta 0:00:00
_____ 21.1/21.1 MB 55.8 MB/s eta 0:00:00
_____ 341.4/341.4 kB 33.7 MB/s eta 0:00:00
_____ 3.4/3.4 MB 88.8 MB/s eta 0:00:00
_____ 1.3/1.3 MB 52.9 MB/s eta 0:00:00
_____ 130.2/130.2 kB 13.2 MB/s eta 0:00:00
_____ 86.8/86.8 kB 9.0 MB/s eta 0:00:00

```

```
Building wheel for pypika (pyproject.toml) ... done
```

```
# Import all the required Libraries
```

```

import pdfplumber
from pathlib import Path
import pandas as pd
from operator import itemgetter
import json
import tiktoken
import openai
import chromadb

```

```
# Mount Google Drive
```

```

from google.colab import drive
drive.mount('/content/drive', force_remount=True)

```

```
Mounted at /content/drive
```

2. Read, Process, and Chunk the PDF Files

We will be using [pdfplumber](#) to read and process the PDF files.

pdfplumber allows for better parsing of the PDF file as it can read various elements of the PDF apart from the plain text, such as, tables, images, etc. It also offers wide functionalities and visual debugging features to help with advanced preprocessing as well.

```
# Define the path of the PDF
```

```
single_pdf_path = '/content/drive/MyDrive/GenAI/project-Mr-HelpMate-AI/insurance-document'
```

2.1 Reading a single PDF file and exploring it through pdfplumber

```
# Function to check whether a word is present in a table or not for segregation of regular text and tables
```

```
def check_bboxes(word, table_bbox):
```

```
    # Check whether word is inside a table bbox.
```

```
    l = word['x0'], word['top'], word['x1'], word['bottom']
```

```
    r = table_bbox
```

```
    return l[0] > r[0] and l[1] > r[1] and l[2] < r[2] and l[3] < r[3]
```

```
# Function to extract text from a PDF file.
```

```
# 1. Declare a variable p to store the iteration of the loop that will help us store page numbers alongside the text
```

```
# 2. Declare an empty list 'full_text' to store all the text files
```

```
# 3. Use pdfplumber to open the pdf pages one by one
```

```
# 4. Find the tables and their locations in the page
```

```
# 5. Extract the text from the tables in the variable 'tables'
```

```
# 6. Extract the regular words by calling the function check_bboxes() and checking whether words are present in the table or not
```

```
# 7. Use the cluster_objects utility to cluster non-table and table words together so that they retain the same chronology as in the original
```

```
# 8. Declare an empty list 'lines' to store the page text
```

```
# 9. If a text element is present in the cluster, append it to 'lines', else if a table element is present, append the table
```

```
# 10. Append the page number and all lines to full_text, and increment 'p'
```

```
# 11. When the function has iterated over all pages, return the 'full_text' list
```

```
def extract_text_from_pdf(pdf_path):
```

```
    p = 0
```

```
    full_text = []
```

```
    with pdfplumber.open(pdf_path) as pdf:
```

```
        for page in pdf.pages:
```

```
            page_no = f"Page {p+1}"
```

```
            text = page.extract_text()
```

```
            tables = page.find_tables()
```

```
            table_bboxes = [i.bbox for i in tables]
```

```
            tables = [{ 'table': i.extract(), 'top': i.bbox[1]} for i in tables]
```

```
            non_table_words = [word for word in page.extract_words() if not any(
```

```
                [check_bboxes(word, table_bbox) for table_bbox in table_bboxes])]
```

```
            lines = []
```

```
            for cluster in pdfplumber.utils.cluster_objects(non_table_words + tables, itemgetter('top'), tolerance=5):
```

```
                if 'text' in cluster[0]:
```

```
                    try:
```

```
                        lines.append(' '.join([i['text'] for i in cluster]))
```

```
                    except KeyError:
```

```
                        pass
```

```
                elif 'table' in cluster[0]:
```

```
                    lines.append(json.dumps(cluster[0]['table']))
```

```
            full_text.append([page_no, " ".join(lines)])
```

```
            p +=1
```

```
    return full_text
```

Now that we have defined the function for extracting the text and tables from a PDF, let's iterate and call this function for all the PDFs in our drive and store them in a list.

```
pdf_path = "/content/drive/MyDrive/GenAI/project-Mr-HelpMate-AI/insurance-document/Principal-Sample-Life-Insurance-Policy.pdf"

# Initialize an empty list to store the extracted texts and document names
data = []

# Process the PDF file
print(f"...Processing {pdf_path}")

# Call the function to extract the text from the PDF
extracted_text = extract_text_from_pdf(pdf_path)

# Convert the extracted list to a PDF, and add a column to store document names
extracted_text_df = pd.DataFrame(extracted_text, columns=['Page No.', 'Page_Text'])

# Append the extracted text and document name to the list
data.append(extracted_text_df)




# Print a message to indicate progress
print(f"Finished processing {pdf_path}")

...Processing /content/drive/MyDrive/GenAI/project-Mr-HelpMate-AI/insurance-document/Principal-Sample-Life-Insurance-Policy.pdf
Finished processing /content/drive/MyDrive/GenAI/project-Mr-HelpMate-AI/insurance-document/Principal-Sample-Life-Insurance-Policy.pdf

# Concatenate all the DFs in the list 'data' together

insurance_pdf_data = pd.concat(data, ignore_index=True)
```

insurance_pdf_data

	Page No.	Page_Text	
0	Page 1	DOROTHEA GLAUSE S655 RHODE ISLAND JOHN DOE 01/...	
1	Page 2	This page left blank intentionally	
2	Page 3	POLICY RIDER GROUP INSURANCE POLICY NO: S655 C...	
3	Page 4	This page left blank intentionally	
4	Page 5	PRINCIPAL LIFE INSURANCE COMPANY (called The P...	
...	
59	Page 60	I f a Dependent who was insured dies during th...	
60	Page 61	Section D - Claim Procedures Article 1 - Notic...	
61	Page 62	A claimant may request an appeal of a claim de...	
62	Page 63	This page left blank intentionally	
63	Page 64	Principal Life Insurance Company Des Moines, I...	

64 rows × 2 columns

Next steps:

[Generate code with insurance_pdf_data](#)

 [View recommended plots](#)

Let's also check the length of all the texts as there might be some empty pages or pages with very few words that we can drop

```
insurance_pdf_data['Text_Length'] = insurance_pdf_data['Page_Text'].apply(lambda x: len(x.split(' ')))

insurance_pdf_data['Text_Length']

0      30
1       5
2     230
3       5
4     110
...
59     285
60     418
61     322
62       5
63       8
Name: Text_Length, Length: 64, dtype: int64
```

```
# Retain only the rows with a text length of at least 10
```

```
insurance_pdf_data = insurance_pdf_data.loc[insurance_pdf_data['Text_Length'] >= 10]  
insurance_pdf_data
```



	Page No.	Page_Text	Text_Length	
0	Page 1	DOROTHEA GLAUSE S655 RHODE ISLAND JOHN DOE 01/...	30	
2	Page 3	POLICY RIDER GROUP INSURANCE POLICY NO: S655 C...	230	
4	Page 5	PRINCIPAL LIFE INSURANCE COMPANY (called The P...	110	
5	Page 6	TABLE OF CONTENTS PART I - DEFINITIONS PART II...	153	
6	Page 7	Section A – Eligibility Member Life Insurance ...	176	
7	Page 8	Section A - Member Life Insurance Schedule of ...	171	
8	Page 9	PART I - DEFINITIONS When used in this Group ...	387	
9	Page 10	T he legally recognized union of two eligible ...	251	
10	Page 11	(2) has been placed with the Member or spouse ...	299	
11	Page 12	An institution that is licensed as a Hospital ...	352	
12	Page 13	a . A licensed Doctor of Medicine (M.D.) or Os...	260	
13	Page 14	c . end stage renal failure; or d. acquired im...	316	
14	Page 15	A record which is on or transmitted by paper o...	36	
15	Page 16	PART II - POLICY ADMINISTRATION Section A - Co...	325	
16	Page 17	a. be actively engaged in business for profit ...	280	
17	Page 18	c . a copy of the form which contains the stat...	291	
18	Page 19	T he Principal has complete discretion to cons...	150	
19	Page 20	Section B - Premiums Article 1 - Payment Respo...	321	
20	Page 21	b . on any date the definition of Member or De...	370	
21	Page 22	The number of Members insured for Dependent Li...	222	
22	Page 23	Section C - Policy Termination Article 1 - Fai...	345	
23	Page 24	T he Principal may terminate the Policyholder'...	113	
24	Page 25	Section D - Policy Renewal Article 1 - Renewal...	79	
25	Page 26	PART III - INDIVIDUAL REQUIREMENTS AND RIGHTS ...	250	
26	Page 27	I f a Member's Dependent is employed and is co...	87	
27	Page 28	Section B - Effective Dates Article 1 - Member...	367	
28	Page 29	Insurance for which Proof of Good Health is re...	408	
29	Page 30	(6) If, on the date a Member becomes eligible ...	462	
30	Page 31	Scheduled Benefit in force for the Member befo...	449	
31	Page 32	(1) marriage or establishment of a Civil Union...	429	
32	Page 33	a . In no event will Dependent Life Insurance ...	460	
33	Page 34	provided The Principal has been notified of th...	94	
34	Page 35	Section C - Individual Terminations Article 1 ...	244	
35	Page 36	A Member's insurance under this Group Policy f...	333	
36	Page 37	b. a business assignment; or c. full-time stud...	124	
37	Page 38	Section D - Continuation Article 1 - Member Li...	317	
38	Page 39	(1) the child is incapable of self-support as ...	206	
39	Page 40	Section E - Reinstatement Article 1 - Reinstat...	322	
40	Page 41	I f coverage for a Member or Dependent termina...	253	
41	Page 42	Section F - Individual Purchase Rights Article...	376	
42	Page 43	Any individual policy issued will then be in f...	392	
43	Page 44	(4) Premium will be based on the Dependent's a...	359	
44	Page 45	(1) If termination is as described in b. (1) a...	179	
45	Page 46	PART IV - BENEFITS Section A - Member Life Ins...	289	
46	Page 47	M ember's death, the Death Benefits Payable ma...	391	
47	Page 48	c . If a beneficiary dies at the same time or ...	420	

48	Page 49	Payment of benefits will be subject to the Ben...	380
49	Page 50	The Principal may require that a ADL Disabled ...	414
50	Page 51	Coverage During Disability will cease on the e...	273
51	Page 52	(1) only one Accelerated Benefit payment will ...	215
52	Page 53	Section B - Member Accidental Death and Dismem...	287
53	Page 54	f . claim requirements listed in PART IV, Sect...	368
54	Page 55	Exposure Exposure to the elements will be pres...	327
55	Page 56	If a Member sustains an injury, and as a resul...	307
56	Page 57	% of Scheduled Covered Loss Benefit Loss of Sp...	321
57	Page 58	a. willful self-injury or self-destruction, wh...	214
58	Page 59	Section C - Dependent Life Insurance Article 1...	240
59	Page 60	I f a Dependent who was insured dies during th...	285
60	Page 61	Section D - Claim Procedures Article 1 - Notic...	418
61	Page 62	A claimant may request an appeal of a claim de...	322

Next steps:

[Generate code with insurance_pdf_data](#)[View recommended plots](#)

Store the metadata for each page in a separate column






insurance_pdf_data['Metadata'] = insurance_pdf_data.apply(lambda x: {'Page_No.': x['Page No.']], axis=1)

```
<ipython-input-65-72c705262e1f>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
insurance_pdf_data['Metadata'] = insurance_pdf_data.apply(lambda x: {'Page_No.': x['Page No.']], axis=1)
```

insurance_pdf_data

	Page No.	Page_Text	Text_Length	Metadata	
0	Page 1	DOROTHEA GLAUSE S655 RHODE ISLAND JOHN DOE 01/...	30	{'Page_No.': 'Page 1'}	  
2	Page 3	POLICY RIDER GROUP INSURANCE POLICY NO: S655 C...	230	{'Page_No.': 'Page 3'}	
4	Page 5	PRINCIPAL LIFE INSURANCE COMPANY (called The P...	110	{'Page_No.': 'Page 5'}	
5	Page 6	TABLE OF CONTENTS PART I - DEFINITIONS PART II...	153	{'Page_No.': 'Page 6'}	  
6	Page 7	Section A – Eligibility Member Life Insurance ...	176	{'Page_No.': 'Page 7'}	
7	Page 8	Section A - Member Life Insurance Schedule of ...	171	{'Page_No.': 'Page 8'}	
8	Page 9	PART I - DEFINITIONS When used in this Group ...	387	{'Page_No.': 'Page 9'}	  
9	Page 10	T he legally recognized union of two eligible ...	251	{'Page_No.': 'Page 10'}	
10	Page 11	(2) has been placed with the Member or spouse ...	299	{'Page_No.': 'Page 11'}	
11	Page 12	An institution that is licensed as a Hospital ...	352	{'Page_No.': 'Page 12'}	  
12	Page 13	a . A licensed Doctor of Medicine (M.D.) or Os...	260	{'Page_No.': 'Page 13'}	
13	Page 14	c . end stage renal failure; or d. acquired im...	316	{'Page_No.': 'Page 14'}	
14	Page 15	A record which is on or transmitted by paper o...	36	{'Page_No.': 'Page 15'}	  
15	Page 16	PART II - POLICY ADMINISTRATION Section A - Co...	325	{'Page_No.': 'Page 16'}	
16	Page 17	a. be actively engaged in business for profit ...	280	{'Page_No.': 'Page 17'}	
17	Page 18	c . a copy of the form which contains the stat...	291	{'Page_No.': 'Page 18'}	  
18	Page 19	T he Principal has complete discretion to cons...	150	{'Page_No.': 'Page 19'}	
19	Page 20	Section B - Premiums Article 1 - Payment Respo...	321	{'Page_No.': 'Page 20'}	
20	Page 21	b . on any date the definition of Member or De...	370	{'Page_No.': 'Page 21'}	  
21	Page 22	The number of Members insured for Dependent Li...	222	{'Page_No.': 'Page 22'}	
22	Page 23	Section C - Policy Termination Article 1 - Fai...	345	{'Page_No.': 'Page 23'}	
23	Page 24	T he Principal may terminate the Policyholder'...	113	{'Page_No.': 'Page 24'}	  
24	Page 25	Section D - Policy Renewal Article 1 - Renewal...	79	{'Page_No.': 'Page 25'}	
25	Page 26	PART III - INDIVIDUAL REQUIREMENTS AND RIGHTS ...	250	{'Page_No.': 'Page 26'}	
26	Page 27	I f a Member's Dependent is employed and is co...	87	{'Page_No.': 'Page 27'}	  
27	Page 28	Section B - Effective Dates Article 1 - Member...	367	{'Page_No.': 'Page 28'}	
28	Page 29	Insurance for which Proof of Good Health is re...	408	{'Page_No.': 'Page 29'}	
29	Page 30	(6) If, on the date a Member becomes eligible ...	462	{'Page_No.': 'Page 30'}	  
30	Page 31	Scheduled Benefit in force for the Member befo...	449	{'Page_No.': 'Page 31'}	
31	Page 32	(1) marriage or establishment of a Civil Union...	429	{'Page_No.': 'Page 32'}	
32	Page 33	a . In no event will Dependent Life Insurance ...	460	{'Page_No.': 'Page 33'}	  
33	Page 34	provided The Principal has been notified of th...	94	{'Page_No.': 'Page 34'}	
34	Page 35	Section C - Individual Terminations Article 1 ...	244	{'Page_No.': 'Page 35'}	
35	Page 36	A Member's insurance under this Group Policy f...	333	{'Page_No.': 'Page 36'}	  
36	Page 37	b. a business assignment; or c. full-time stud...	124	{'Page_No.': 'Page 37'}	
37	Page 38	Section D - Continuation Article 1 - Member Li...	317	{'Page_No.': 'Page 38'}	
38	Page 39	(1) the child is incapable of self-support as ...	206	{'Page_No.': 'Page 39'}	  
39	Page 40	Section E - Reinstatement Article 1 - Reinstat...	322	{'Page_No.': 'Page 40'}	
40	Page 41	I f coverage for a Member or Dependent termina...	253	{'Page_No.': 'Page 41'}	
41	Page 42	Section F - Individual Purchase Rights Article...	376	{'Page_No.': 'Page 42'}	  
42	Page 43	Any individual policy issued will then be in f...	392	{'Page_No.': 'Page 43'}	
43	Page 44	(4) Premium will be based on the Dependent's a...	359	{'Page_No.': 'Page 44'}	
44	Page 45	(1) If termination is as described in b. (1) a...	179	{'Page_No.': 'Page 45'}	  
45	Page 46	PART IV - BENEFITS Section A - Member Life Ins...	289	{'Page_No.': 'Page 46'}	
46	Page 47	M ember's death, the Death Benefits Payable ma...	391	{'Page_No.': 'Page 47'}	
47	Page 48	c . If a beneficiary dies at the same time or ...	420	{'Page_No.': 'Page 48'}	

48	Page 49	Payment of benefits will be subject to the Ben...	380	{'Page_No.': 'Page 49'}
49	Page 50	The Principal may require that a ADL Disabled ...	414	{'Page_No.': 'Page 50'}
50	Page 51	Coverage During Disability will cease on the e...	273	{'Page_No.': 'Page 51'}
51	Page 52	(1) only one Accelerated Benefit payment will ...	215	{'Page_No.': 'Page 52'}
52	Page 53	Section B - Member Accidental Death and Dismem...	287	{'Page_No.': 'Page 53'}
53	Page 54	f . claim requirements listed in PART IV, Sect...	368	{'Page_No.': 'Page 54'}
54	Page 55	Exposure Exposure to the elements will be pres...	327	{'Page_No.': 'Page 55'}
55	Page 56	If a Member sustains an injury, and as a resul...	307	{'Page_No.': 'Page 56'}
56	Page 57	% of Scheduled Covered Loss Benefit Loss of Sp...	321	{'Page_No.': 'Page 57'}
57	Page 58	a. willful self-injury or self-destruction, wh...	214	{'Page_No.': 'Page 58'}
58	Page 59	Section C - Dependent Life Insurance Article 1...	240	{'Page_No.': 'Page 59'}
59	Page 60	I f a Dependent who was insured dies during th...	285	{'Page_No.': 'Page 60'}
60	Page 61	Section D - Claim Procedures Article 1 - Notic...	418	{'Page_No.': 'Page 61'}
61	Page 62	A claimant may request an appeal of a claim de...	322	{'Page_No.': 'Page 62'}

Next steps:

[Generate code with insurance_pdf_data](#)[View recommended plots](#)

This concludes the chunking aspect also, as we can see that mostly the pages contain few hundred words, maximum going upto 1000. So, we don't need to chunk the documents further; we can perform the embeddings on individual pages. This strategy makes sense for 2 reasons:

1. The way insurance documents are generally structured, you will not have a lot of extraneous information in a page, and all the text pieces in that page will likely be interrelated.
2. We want to have larger chunk sizes to be able to pass appropriate context to the LLM during the generation layer.

✓ 3. Generate and Store Embeddings using OpenAI and ChromaDB

In this section, we will embed the pages in the dataframe through OpenAI's `text-embedding-ada-002` model, and store them in a ChromaDB collection.

```
# Set the API key
```

```
with open("/content/drive/MyDrive/GenAI/project-Mr-HelpMate-AI/OpenAI_API_Key.txt", "r") as f:
    openai.api_key = ' '.join(f.readlines())
```

```
# Import the OpenAI Embedding Function into chroma
```

```
from chromadb.utils.embedding_functions import OpenAIEmbeddingFunction
```

```
# Define the path where chroma collections will be stored
```

```
chroma_data_path = '/content/drive/MyDrive/GenAI/project-Mr-HelpMate-AI/ChromaDB_Data'
```

```
import chromadb
```

```
# Call PersistentClient()
```

```
client = chromadb.PersistentClient()
```

```
# Set up the embedding function using the OpenAI embedding model
```

```
model = "text-embedding-ada-002"
embedding_function = OpenAIEmbeddingFunction(api_key=openai.api_key, model_name=model)
```

```
# Initialise a collection in chroma and pass the embedding_function to it so that it used OpenAI embeddings to embed the documents
insurance_collection = client.get_or_create_collection(name='RAG_on_Insurance', embedding_function=embedding_function)
```



```
# Convert the page text and metadata from your dataframe to lists to be able to pass it to chroma

documents_list = insurance_pdf_data["Page_Text"].tolist()
metadata_list = insurance_pdf_data['Metadata'].tolist()

# Add the documents and metadata to the collection alongwith generic integer IDs. You can also feed the metadata information as IDs by combi

insurance_collection.add(
    documents= documents_list,
    ids = [str(i) for i in range(0, len(documents_list))],
    metadatas = metadata_list
)

cache_collection = client.get_or_create_collection(name='Insurance_Cache', embedding_function=embedding_function)

cache_collection.peek()

{'ids': [],
 'embeddings': [],
 'metadatas': [],
 'documents': [],
 'uris': None,
 'data': None}
```

✓ 4. Semantic Search with Cache

In this section, we will perform a semantic search of a query in the collections embeddings to get several top semantically similar results.

```
# Read the user query
```

```
query = input()
```

What provisions may allow for a longer reinstatement period for an approved leave of absence taken in accordance with the Uniformed Ser

```
# Search the Cache collection first
```

```
# Query the collection against the user query and return the top 20 results
```

```
cache_results = cache_collection.query(
    query_texts=query,
    n_results=1
)
```

```
cache_results
```

```
{'ids': [['How is the peroid of time during which a reinstated Member's insurance was not in force treated for the purpose of
determining the length of continuous coverage under the Group Polocy?']],
 'distances': [[0.3391200096784594]],
 'metadatas': [[{'distances0': '0.22027961825216222',
 'distances1': '0.2699016759608078',
 'distances2': '0.3192471758174104',
 'distances3': '0.32401942839566994',
 'distances4': '0.328616716784959',
 'distances5': '0.33030814255508284',
 'distances6': '0.33096864446296964',
 'distances7': '0.3369859985974198',
 'distances8': '0.33744009745823667',
 'distances9': '0.3384475398423276',
 'documents0': "Section E - Reinstatement Article 1 - Reinstatement A Member's terminated insurance will be reinstated if: a.
insurance ceased because of layoff or approved leave of absence; and b. the Member returns to Active Work for the Policyholder
within six months of the date insurance ceased. The Member's reinstated insurance will be in force on the date of return to work.
However, the Actively at Work and Period of Limited Activity provisions discussed in PART III, Section B, will apply. Also, Proof
of Good Health will be required to place in force any Scheduled Benefit that would have been subject to Proof of Good Health had
the Member remained continuously insured. Only the period of time during which a Member is actually insured will be included in
determining the length of his or her continuous coverage under this Group Policy. For this purpose the period of time during which
a reinstated Member's insurance was not in force: a. will not be considered an interruption of continuous coverage; and b. will not
be used to satisfy any provision of this Group Policy which pertains to a period of continuous coverage. In addition, a longer
reinstatement period may be allowed for an approved leave of absence taken in accordance with the provisions of the federal law
regarding the Uniformed Services Employment and Reemployment Rights Act of 1994 (USERRA). Article 2 - Federal Required Family and
Medical Leave Act (FMLA) A Member's terminated insurance may be reinstated in accordance with the provisions of the Federal Family
and Medical Leave Act (FMLA), subject to the Actively at Work and Period of Limited Activity provision discussed in PART III,
Section B. Article 3 - Reinstatement of Coverage for a Member or Dependent When Coverage Ends due to Living Outside of the United
States This policy has been updated effective January 1, 2014 PART III - INDIVIDUAL REQUIREMENTS AND RIGHTS GC 6010 Section E -
```

Reinstatement, Page 1",

'documents1': 'I f coverage for a Member or Dependent terminates because the person is outside of the United States as discussed in PART III, Section C, Article 5, the Member or Dependent may become eligible again for coverage under this Group Policy, but only if: a. the Member or Dependent return to the United States within six months of the date on which coverage terminated because the person is outside of the United States; and b. in the case of a Member, the Member returns to Active Work in the United States for the Policyholder for a period of at least 30 consecutive days. The Member will be eligible for coverage on the day immediately following completion of the 30 consecutive days of Active Work; and c. in the case of the Dependent, he or she remains in the United States for 30 consecutive days. If the Dependent does so, he or she will be eligible for reinstatement of coverage on the day after completion of the 30 consecutive days of residence. The reinstated coverage will be on the same basis as that being provided on the date coverage is reinstated. However, any restrictions on this coverage that were in effect before reinstatement will continue to apply. If the Member or Dependent does not complete the 30 consecutive days of residence, the coverage for such person will not be reinstated. This policy has been updated effective January 1, 2014 PART III - INDIVIDUAL REQUIREMENTS AND RIGHTS GC 6010 Section E - Reinstatement, Page 2',

'documents2': "a. be actively engaged in business for profit within the meaning of the Internal Revenue Code, or be established as a legitimate nonprofit corporation within the meaning of the Internal Revenue Code; and b. make at least the level of premium contributions required for insurance on its eligible Members. The Policyholder must: (1) contribute at least 50% of the required premium for all Members (including disabled Members, if any); and c. if the Member is to contribute part of the premium, maintain the following participation percentages with respect to eligible employees and Dependents, excluding those for whom Proof of Good Health is not satisfactory to The Principal: (1) Employees: - at least 75% of all eligible employees must enroll; (2) Dependents: - maintain a Dependent participation of at least 75% of eligible Dependents; and d. if the Member is to contribute no part of the premium, 100% of eligible employees and Dependents must enroll. Article 4 - Policy Incontestability In the absence of fraud, after this Group Policy has been in force two years, The Principal may not contest its validity except for nonpayment of premium. Article 5 - Individual Incontestability All statements made by any individual insured under this Group Policy will be representations and not warranties. In the absence of fraud, these statements may not be used to contest an insured person's insurance unless: a. the insured person's insurance has been in force for less than two years during the insured's lifetime; and b. the statement is in Written form Signed by the insured person; and This policy has been updated effective January 1, 2014 PART II - POLICY ADMINISTRATION GC 6003 Section A - Contract, Page 2",

'documents3': "Section D - Continuation Article 1 - Member Life Insurance a. Sickness or Injury (Other Than ADL Disability or Total Disability) If Active Work ends because a Member is sick or injured but not ADL Disabled or Totally Disabled, insurance for

```
results = insurance_collection.query(
query_texts=query,
n_results=10
)
```

```

# Implementing Cache in Semantic Search

# Set a threshold for cache search
threshold = 0.2

ids = []
documents = []
distances = []
metadatas = []
results_df = pd.DataFrame()

# If the distance is greater than the threshold, then return the results from the main collection.

if cache_results['distances'][0] == [] or cache_results['distances'][0][0] > threshold:
    # Query the collection against the user query and return the top 10 results
    results = insurance_collection.query(
        query_texts=query,
        n_results=10
    )

    # Store the query in cache_collection as document w.r.t to ChromaDB so that it can be embedded and searched against later
    # Store retrieved text, ids, distances and metadatas in cache_collection as metadatas, so that they can be fetched easily if a query i
    Keys = []
    Values = []

    for key, val in results.items():
        if val is None:
            continue
        for i in range(10):
            Keys.append(str(key)+str(i))
            Values.append(str(val[0][i]))

    cache_collection.add(
        documents= [query],
        ids = [query], # Or if you want to assign integers as IDs 0,1,2,.., then you can use "len(cache_results['documents'])" as will re
        metadatas = dict(zip(Keys, Values))
    )

    print("Not found in cache. Found in main collection.")

    result_dict = {'Metadatas': results['metadatas'][0], 'Documents': results['documents'][0], 'Distances': results['distances'][0], "IDs"
    results_df = pd.DataFrame.from_dict(result_dict)
    results_df

# If the distance is, however, less than the threshold, you can return the results from cache

elif cache_results['distances'][0][0] <= threshold:
    cache_result_dict = cache_results['metadatas'][0][0]

    # Loop through each inner list and then through the dictionary
    for key, value in cache_result_dict.items():
        if 'ids' in key:
            ids.append(value)
        elif 'documents' in key:
            documents.append(value)
        elif 'distances' in key:
            distances.append(value)
        elif 'metadatas' in key:
            metadatas.append(value)

    print("Found in cache!")

    # Create a DataFrame
    results_df = pd.DataFrame({
        'IDs': ids,
        'Documents': documents,
        'Distances': distances,
        'Metadatas': metadatas
    })

    Not found in cache. Found in main collection.

```

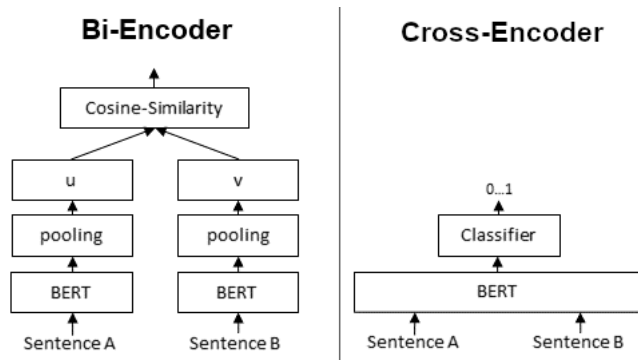
results_df

	Metadatas	Documents	Distances	IDs	
0	{'Page_No.': 'Page 40'} Section E - Reinstatement Article 1 - Reinstat...		0.263336	37	
1	{'Page_No.': 'Page 41'} If coverage for a Member or Dependent termina...		0.353420	38	
2	{'Page_No.': 'Page 37'} b. a business assignment; or c. full-time stud...		0.377362	34	
3	{'Page_No.': 'Page 38'} Section D - Continuation Article 1 - Member Li...		0.417215	35	
4	{'Page_No.': 'Page 61'} Section D - Claim Procedures Article 1 - Notic...		0.424813	58	
5	{'Page_No.': 'Page 62'} A claimant may request an appeal of a claim de...		0.431366	59	
6	{'Page_No.': 'Page 27'} If a Member's Dependent is employed and is co...		0.439407	24	
7	{'Page_No.': 'Page 28'} Section B - Effective Dates Article 1 - Member...		0.442143	25	
8	{'Page_No.': 'Page 54'} f. claim requirements listed in PART IV, Sect...		0.442198	51	
9	{'Page_No.': 'Page 24'} T he Principal may terminate the Policyholder'...		0.448250	21	

Next steps: [Generate code with results_df](#) [View recommended plots](#)

5. Re-Ranking with a Cross Encoder

Re-ranking the results obtained from your semantic search can sometime significantly improve the relevance of the retrieved results. This is often done by passing the query paired with each of the retrieved responses into a cross-encoder to score the relevance of the response w.r.t. the query.



```
# Import the CrossEncoder library from sentence_transformers
```

```
from sentence_transformers import CrossEncoder, util
```

```
# Initialise the cross encoder model
```

```
cross_encoder = CrossEncoder('cross-encoder/ms-marco-MiniLM-L-6-v2')
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
The secret 'HF_TOKEN' does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public model
warnings.warn(
config.json: 100% 794/794 [00:00<00:00, 44.7kB/s]
pytorch_model.bin: 100% 90.9M/90.9M [00:00<00:00, 172MB/s]
tokenizer_config.json: 100% 316/316 [00:00<00:00, 10.5kB/s]
vocab.txt: 100% 232k/232k [00:00<00:00, 3.62MB/s]
special_tokens_map.json: 100% 112/112 [00:00<00:00, 4.87kB/s]
```