1. Install and Import the Required Libraries

```
# Install all the required libraries

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

```
# 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

1. List item
2. List item

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 functionaties 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 in 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}"
```

```
            page_no = "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
```

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-13-72c705262e1f>: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-cc
  insurance_pdf_data['Metadata'] = insurance_pdf_data.apply(lambda x: {'Page_No.': x['Page No.']}, axis=1)
```

```
insurance_pdf_data
```

	42	Article...		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
)
```

```
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 2
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 3
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 4
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 5
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 6
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 7
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 8
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 9
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 10
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 11
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 12
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 13
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 14
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 15
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 16
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 17
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 18
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 19
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 20
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 21
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 22
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 23
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 24
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 25
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 26
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 27
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 28
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 29
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 30
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 31
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 32
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 33
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 34
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 35
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 36
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 37
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 38
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 39
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 40
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 41
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 42
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 43
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 44
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 45
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 46
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 47
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 48
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 49
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 50
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 51
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 52
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 53
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 54
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 55
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 56
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 57
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 58
WARNING:chromadb.segment.impl.vector.local_persistent_hnsw:Add of existing embedding ID: 59
```

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


cache_collection.peek()
```

(or approved amount, if applicable) in force if one foot is severed at or above the ankle; or e. 50% of the Scheduled Benefit (or approved amount, if applicable) in force if the sight of one eye is permanently lost (For this purpose, vision not correctable to better than 20/200 will be considered loss of sight.); or f. 100% of the Scheduled Benefit (or approved amount, if applicable) in force for more than one of the losses listed in b., d., or e. above. Total payment for all losses under this Article 3 that result from the same accident will not exceed the Scheduled Benefit (or approved amount, if applicable). Payment for loss of life will be to the beneficiary named for Member Life Insurance. Payment will be subject to the Beneficiary, Facility of Payment and Settlement of Proceeds provisions of PART IV, Section A. Payment for all other losses will be to the Member. Disappearance It will be presumed that a Member has lost his or her life if: a. the Member's body has not been found within 365 days after the disappearance of a conveyance in which the Member was an occupant at the time of disappearance; and b. the disappearance of the conveyance was due to its accidental wrecking or sinking; and c. this Group Policy would have covered the injury resulting from the accident. This policy has been updated effective January 1, 2014 PART IV - BENEFITS GC 6015 Section B - Member Accidental Death and Dismemberment Insurance, Page 2",
'documents9': "T he Principal may terminate the Policyholder's coverage on any premium due date if the Policyholder relocates to a state where this Group Policy is not marketed, by giving the Policyholder 31 days advanced notice in Writing. Article 4 - Policyholder Responsibility to Members If this Group Policy terminates for any reason, the Policyholder must: a. notify each Member of the effective date of the termination; and b. refund or otherwise account to each Member all contributions received or withheld from Members for premiums not actually paid to The Principal. This policy has been updated effective January 1, 2014 PART II - POLICY ADMINISTRATION GC 6005 Section C - Policy Termination, Page 2",
'ids0': '37',
'ids1': '38',
'ids2': '34',
'ids3': '35',
'ids4': '58',
'ids5': '59',
'ids6': '24',
'ids7': '25',
'ids8': '51',
'ids9': '21',
'metadatas0': "{'Page_No.': 'Page 40'}",
'metadatas1': "{'Page_No.': 'Page 41'}",
'metadatas2': "{'Page_No.': 'Page 37'}",
'metadatas3': "{'Page_No.': 'Page 38'}",
'metadatas4': "{'Page_No.': 'Page 61'}",
'metadatas5': "{'Page_No.': 'Page 62'}",
'metadatas6': "{'Page_No.': 'Page 27'}",
'metadatas7': "{'Page_No.': 'Page 28'}",
'metadatas8': "{'Page_No.': 'Page 54'}",
'metadatas9': "{'Page_No.': 'Page 24'}"}],
'documents': ["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 Policy?",
"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?",
"Under what four conditions may a member's insurance be continued if Active work ends due to layoffs or approved leave of absence?",
'What are the requirements for placing in force any Scheduled benefit that would have been subject to Proof of Good Health has the member remained continuously insured?',
"What povisions will apply to the member's reinstated insurance when they return to work?",
'What provisions may allow for a longer reinstatement period for an approved leave of absensce taken in accordance with the Uniformed Services Employment and Reemployment Rights Act of 1994 (USERRA)?'],
'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 Serv
```

```
# Searh 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
```

change in their insurance, whichever is applicable. This Actively at Work requirement may also be waived as described below. When insurance under this Group Policy replaces coverage under a Prior Policy, the Active Work requirement may be waived for those Members who: (1) are eligible and enrolled under this Group Policy on its Date of Issue; and (2) were covered under the Prior Policy on the date of its termination. In no event will the Active Work requirement be waived for those Members who, on the date of termination of the Prior Policy, either: (1) had the option, under the terms of the Prior Policy, to convert their coverage under the Prior Policy to an individual policy; or (2) were eligible under the terms of the Prior Policy, to have their premiums waived due to ADL Disability or Total Disability. NOTE: When insurance under this Group Policy replaces coverage under a Prior Policy and the Active Work requirement is waived, any benefits payable will be the lesser of the Scheduled Benefit of this Group Policy or the amount that would have been paid by the Prior Policy had it remained in force. b. Effective Date for Initial Insurance When Proof of Good Health is Required This policy has been updated effective January 1, 2014 PART III - INDIVIDUAL REQUIREMENTS AND RIGHTS GC 6007 Section B - Effective Dates, Page 1",
'documents8': "f . claim requirements listed in PART IV, Section D, must be satisfied; and g. all medical evidence must be satisfactory to The Principal. Article 3 - Benefits Payable If all of the benefit qualifications are met, The Principal will pay: a. 100% of the Scheduled Benefit (or approved amount, if applicable) in force for loss of life; or b. 50% of the Scheduled Benefit (or approved amount, if applicable) in force if one hand is severed at or above the wrist; or c. 25% of the Scheduled Benefit (or approved amount, if applicable) in force for loss of thumb and index finger on the same hand; or d. 50% of the Scheduled Benefit (or approved amount, if applicable) in force if one foot is severed at or above the ankle; or e. 50% of the Scheduled Benefit (or approved amount, if applicable) in force if the sight of one eye is permanently lost (For this purpose, vision not correctable to better than 20/200 will be considered loss of sight.); or f. 100% of the Scheduled Benefit (or approved amount, if applicable) in force for more than one of the losses listed in b., d., or e. above. Total payment for all losses under this Article 3 that result from the same accident will not exceed the Scheduled Benefit (or approved amount, if applicable). Payment for loss of life will be to the beneficiary named for Member Life Insurance. Payment will be subject to the Beneficiary, Facility of Payment and Settlement of Proceeds provisions of PART IV, Section A. Payment for all other losses will be to the Member. Disappearance It will be presumed that a Member has lost his or her life if: a. the Member's body has not been found within 365 days after the disappearance of a conveyance in which the Member was an occupant at the time of disappearance; and b. the disappearance of the conveyance was due to its accidental wrecking or sinking; and c. this Group Policy would have covered the injury resulting from the accident. This policy has been updated effective January 1, 2014 PART IV - BENEFITS GC 6015 Section B - Member Accidental Death and Dismemberment Insurance, Page 2",
'documents9': "T he Principal may terminate the Policyholder's coverage on any premium due date if the Policyholder relocates to a state where this Group Policy is not marketed, by giving the Policyholder 31 days advanced notice in Writing. Article 4 - Policyholder Responsibility to Members If this Group Policy terminates for any reason, the Policyholder must: a. notify each Member of the effective date of the termination; and b. refund or otherwise account to each Member all contributions received or withheld from Members for premiums not actually paid to The Principal. This policy has been updated effective January 1, 2014 PART II - POLICY ADMINISTRATION GC 6005 Section C - Policy Termination, Page 2",
'ids0': '37',
'ids1': '38',
'ids2': '34',
'ids3': '35',
'ids4': '58',
'ids5': '59',
'ids6': '24',
'ids7': '25',
'ids8': '51',
'ids9': '21',
'metadatas0': "{'Page_No.': 'Page 40'}",
'metadatas1': "{'Page_No.': 'Page 41'}",
'metadatas2': "{'Page_No.': 'Page 37'}",
'metadatas3': "{'Page_No.': 'Page 38'}",
'metadatas4': "{'Page_No.': 'Page 61'}",
'metadatas5': "{'Page_No.': 'Page 62'}",
'metadatas6': "{'Page_No.': 'Page 27'}",
'metadatas7': "{'Page_No.': 'Page 28'}",
'metadatas8': "{'Page_No.': 'Page 54'}",
'metadatas9': "{'Page_No.': 'Page 24'}"}]],
'embeddings': None,

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

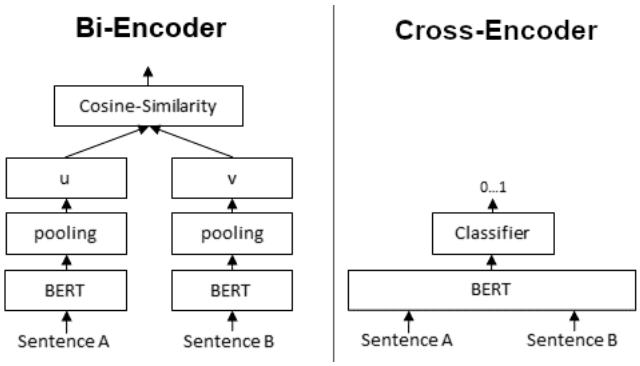
```
# Implementing Cache in Semantic Search

# 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
      })


     Found in cache!


results_df
```

	IDs	Documents	Distances	Metadatas
0	37	Section E - Reinstatement Article 1 - Reinstat...	0.26333645892779955	{'Page_No.': 'Page 40'}
1	38	I f coverage for a Member or Dependent termina...	0.3534203397712997	{'Page_No.': 'Page 41'}
2	34	b. a business assignment; or c. full-time stud...	0.3773623591660169	{'Page_No.': 'Page 37'}
3	35	Section D - Continuation Article 1 - Member Li...	0.41721514261069464	{'Page_No.': 'Page 38'}
4	58	Section D - Claim Procedures Article 1 - Notic...	0.4248132841096369	{'Page_No.': 'Page 61'}
5	59	A claimant may request an appeal of a claim de...	0.43136592712838057	{'Page_No.': 'Page 62'}
6	24	I f a Member's Dependent is employed	0.43940749010830094	{'Page_No.': 'Page

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://huggingface.co/settings/tokens), set it as secre
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 models or datasets.
  warnings.warn(
```

```
# Input (query, response) pairs for each of the top 20 responses received from the semantic search to the cross encoder
# Generate the cross_encoder scores for these pairs

cross_inputs = [[query, response] for response in results_df['Documents']]
cross_rerank_scores = cross_encoder.predict(cross_inputs)


cross_rerank_scores
```