

ATME COLLEGE OF ENGINEERING

13th KM Stone, Bannur Road, Mysore - 560028



A T M E
College of Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING(DATA SCIENCE)

(ACADEMIC YEAR 2024-25)

LABORATORY MANUAL

SUBJECT : GENERATIVE AI

SUB CODE: BAIL657C

SEMESTER: VI-2022 CBCS Scheme

Prepared By by

**Mrs. MADHU NAGARAJ
FACULTY IN-CHARGE**

Approved by

**Dr. ANITHA D B
HOD, CSE- DS**

INSTITUTIONAL MISSION AND VISION

Objectives

- To provide quality education and groom top-notch professionals, entrepreneurs and leaders for different fields of engineering, technology and management.
- To open a Training-R & D-Design-Consultancy cell in each department, gradually introduce doctoral and postdoctoral programs, encourage basic & applied researchin areas of social relevance, and develop the institute as a center of excellence.
- To develop academic, professional and financial alliances with the industry as well asthe academia at national and transnational levels
- To develop academic, professional and financial alliances with the industry as well asthe academia at national and transnational levels.
- To cultivate strong community relationships and involve the students and the staff in local community service.
- To constantly enhance the value of the educational inputs with the participation of students, faculty, parents and industry.

Vision

- Development of academically excellent, culturally vibrant, socially responsible and globally competent human resources.

Mission

- To keep pace with advancements in knowledge and make the students competitive and capable at the global level.
- To create an environment for the students to acquire the right physical, intellectual, emotional and moral foundations and shine as torch bearers of tomorrow's society.
- To strive to attain ever-higher benchmarks of educational excellence.

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING AND ENGINEERING
(DATA SCIENCE &ENGINEERING)**

Vision of the Department

- To impart technical education in the field of data science of excellent quality with a high level of professional competence, social responsibility, and global awareness among the students

Mission

- To impart technical education that is up to date, relevant and makes students competitive and employable at global level
- To provide technical education with a high sense of discipline, social relevance in an intellectually, ethically and socially challenging environment for better tomorrow
- Educate to the global standards with a benchmark of excellence and to kindle the spirit of innovation.

Program Outcomes(PO)

- **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

- **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
- **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1: Develop relevant programming skills to become a successful data scientist

- PSO2: Apply data science concepts and algorithms to solve real world problems of the society
- PSO3: Apply data science techniques in the various domains like agriculture, education healthcare for better society

Program Educational Objectives (PEOs):

PEO1: Develop cutting-edge skills in data science and its related technologies, such as machine learning, predictive analytic, and data engineering.

PEO2: Design and develop data-driven solutions to real-world problems in a business, research, or social environment.

PEO3: Apply data engineering and data visualization techniques to discover, investigate, and interpret data.

PEO4: Demonstrate ethical and responsible data practices in problem solving

PEO5: Integrate fields within computer science, optimization, and statistics to develop better solutions

SL.No	Particulars
1	Program 1: Explore pre-trained word vectors. Explore word relationships using vector arithmetic. Perform arithmetic operations and analyze results.
2	Program 2: Use dimensionality reduction (e.g., PCA or t-SNE) to visualize word embeddings for Q 1. Select 10 words from a specific domain (e.g., sports, technology) and visualize their embeddings. Analyze clusters and relationships. Generate contextually rich outputs using embeddings. Write a program to generate 5 semantically similar words for a given input.
3	Program 3: Train a custom Word2Vec model on a small dataset. Train embeddings on a domain-specific corpus (e.g., legal, medical) and analyze how embeddings capture domain-specific semantics.
4	Program 4: Use word embeddings to improve prompts for Generative AI model. Retrieve similar words using word embeddings. Use the similar words to enrich a GenAI prompt. Use the AI model to generate responses for the original and enriched prompts. Compare the outputs in terms of detail
5	Program 5: Use word embeddings to create meaningful sentences for creative tasks. Retrieve similar words for a seed word. Create a sentence or story using these words as a starting point. Write a program that: Takes a seed word. Generates similar words. Constructs a short paragraph using these words.
6	Program 6: Use a pre-trained Hugging Face model to analyze sentiment in text. Assume a real-world application, Load the sentiment analysis pipeline. Analyze the sentiment by giving sentences to input.
7	Program 7: Summarize long texts using a pre-trained summarization model using Hugging face model. Load the summarization pipeline. Take a passage as input and obtain the summarized text.
8	Program 8: Install langchain, cohore (for key), langchain-community. Get the api key(By logging into Cohere and obtaining the cohore key). Load a text document from your google drive . Create a prompt template to display the output in a particular manner.

9	<p>Program 9:</p> <p>Take the Institution name as input. Use Pydantic to define the schema for the desired output and create a custom output parser. Invoke the Chain and Fetch Results. Extract the below Institution related details from Wikipedia: The founder of the Institution. When it was founded. The current branches in the institution. How many employees are working in it. A brief 4-line summary of the institution.</p>
10	<p>Program 10:</p> <p>Build a chatbot for the Indian Penal Code. We'll start by downloading the official Indian Penal Code document, and then we'll create a chatbot that can interact with it. Users will be able to ask questions about the Indian Penal Code and have a conversation with it.</p>

Syllabus

Generative AI		Semester	6
Course Code	BAIL657C	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	0:0:1:0	SEE Marks	50
Credits	01	Exam Hours	100
Examination type (SEE)	Practical		

Course objectives:

- Understand the principles and concepts behind generative AI models
- Explain the knowledge gained to implement generative models using Prompt design frameworks.
- Apply various Generative AI applications for increasing productivity.
- Develop Large Language Model-based Apps.

SL.NO	Experiments
1.	Explore pre-trained word vectors. Explore word relationships using vector arithmetic. Perform arithmetic operations and analyze results.
2.	Use dimensionality reduction (e.g., PCA or t-SNE) to visualize word embeddings for Q 1. Select 10 words from a specific domain (e.g., sports, technology) and visualize their embeddings. Analyze clusters and relationships. Generate contextually rich outputs using embeddings. Write a program to generate 5 semantically similar words for a given input.
3.	Train a custom Word2Vec model on a small dataset. Train embeddings on a domain-specific corpus (e.g., legal, medical) and analyze how embeddings capture domain-specific semantics.
4.	Use word embeddings to improve prompts for Generative AI model. Retrieve similar words using word embeddings. Use the similar words to enrich a GenAI prompt. Use the AI model to generate responses for the original and enriched prompts. Compare the outputs in terms of detail and relevance.
5.	Use word embeddings to create meaningful sentences for creative tasks. Retrieve similar words for a seed word. Create a sentence or story using these words as a starting point. Write a program that: Takes a seed word. Generates similar words. Constructs a short paragraph using these words.
6.	Use a pre-trained Hugging Face model to analyze sentiment in text. Assume a real-world application, Load the sentiment analysis pipeline. Analyze the sentiment by giving sentences to input.
7.	Summarize long texts using a pre-trained summarization model using Hugging face model. Load the summarization pipeline. Take a passage as input and obtain the summarized text.
8.	Install langchain, cohore (for key), langchain-community. Get the api key(By logging into Cohere and obtaining the cohore key). Load a text document from your google drive . Create a prompt template to display the output in a particular manner.
9.	Take the Institution name as input. Use Pydantic to define the schema for the desired output and create a custom output parser. Invoke the Chain and Fetch Results. Extract the below Institution related details from Wikipedia: The founder of the Institution. When it was founded. The current branches in the institution . How many employees are working in it. A brief 4-line summary of the institution.
10	Build a chatbot for the Indian Penal Code. We'll start by downloading the official Indian Penal Code document, and then we'll create a chatbot that can interact with it. Users will be able to ask questions about the Indian Penal Code and have a conversation with it.

Course outcomes (Course Skill Set):

At the end of the course the student will be able to:

- a. Develop the ability to explore and analyze word embeddings, perform vector arithmetic to investigate word relationships, visualize embeddings using dimensionality reduction techniques
- b. Apply prompt engineering skills to real-world scenarios, such as information retrieval, text generation.
- c. Utilize pre-trained Hugging Face models for real-world applications, including sentiment analysis and text summarization.
- d. Apply different architectures used in large language models, such as transformers, and understand their advantages and limitations.

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

Continuous Internal Evaluation (CIE):

CIE marks for the practical course are **50 Marks**.

The split-up of CIE marks for record/ journal and test are in the ratio **60:40**.

1. Each experiment is to be evaluated for conduction with an observation sheet and record write-up. Rubrics for the evaluation of the journal/write-up for hardware/software experiments are designed by the faculty who is handling the laboratory session and are made known to students at the beginning of the practical session.
2. Record should contain all the specified experiments in the syllabus and each experiment write-up will be evaluated for 10 marks.
3. Total marks scored by the students are scaled down to **30 marks** (60% of maximum marks).
4. Weightage to be given for neatness and submission of record/write-up on time.
5. Department shall conduct a test of 100 marks after the completion of all the experiments listed in the syllabus.
6. In a test, test write-up, conduction of experiment, acceptable result, and procedural knowledge will carry a weightage of 60% and the rest 40% for viva-voce.
7. The suitable rubrics can be designed to evaluate each student's performance and learning ability.
8. The marks scored shall be scaled down to **20 marks** (40% of the maximum marks).

The Sum of scaled-down marks scored in the report write-up/journal and marks of a test is the total CIE marks scored by the student.

Semester End Evaluation (SEE):

- SEE marks for the practical course are 50 Marks.
- SEE shall be conducted jointly by the two examiners of the same institute, examiners are appointed by the Head of the Institute.
- The examination schedule and names of examiners are informed to the university before the conduction of the examination. These practical examinations are to be conducted between the schedule mentioned in the academic calendar of the University.
- All laboratory experiments are to be included for practical examination.
- (Rubrics) Breakup of marks and the instructions printed on the cover page of the answer script to be strictly adhered to by the examiners. OR based on the course requirement evaluation rubrics shall be decided jointly by examiners.
- Students can pick one question (experiment) from the questions lot prepared by the examiners jointly.
- Evaluation of test write-up/ conduction procedure and result/viva will be conducted jointly by examiners. General rubrics suggested for SEE are mentioned here, writeup-20%, Conduction procedure and result in - 60%, Viva-voce 20% of maximum marks. SEE for practical shall be evaluated for 100 marks and scored marks shall be scaled down to 50 marks (however, based on course type, rubrics shall be decided by the examiners)

Change of experiment is allowed only once and 15% of Marks allotted to the procedure part are to be made zero.

The minimum duration of SEE is 02 hours

Suggested Learning Resources:**Books:**

1. Modern Generative AI with ChatGPT and OpenAI Models: Leverage the Capabilities of OpenAI's LLM for Productivity and Innovation with GPT3 and GPT4, by Valentina Alto, Packt Publishing Ltd, 2023.
2. Generative AI for Cloud Solutions: Architect modern AI LLMs in secure, scalable, and ethical cloud environments, by Paul Singh, Anurag Karuparti ,Packt Publishing Ltd, 2024.

Web links and Video Lectures (e-Resources):

- https://www.w3schools.com/gen_ai/index.php
- <https://youtu.be/eTPiL3DF27U>
- <https://youtu.be/je6AlVeGOV0>
- <https://youtu.be/RLVqsA8ns6k>
- <https://youtu.be/0SAKM7wiC-A>
- https://youtu.be/28_9xMyrdjg
- <https://youtu.be/8iuiz-c-EBw>
- <https://youtu.be/7oQ8VtEKcgE>
- <https://youtu.be/seXp0VWWZV0>

About Gen AI

Generative AI leverages deep learning techniques such as neural networks and transformers to create new data instances that resemble training data. This field has seen rapid advancements with the rise of generative adversarial networks (GANs) and diffusion models. With the explosion of large-scale models such as OpenAI's GPT series and Google's Bard, AI is reshaping industries by enabling automated creativity and innovation.

Benefits of the Course

1. **Comprehensive Hands-on Learning** – The course provides hands-on experience with generative models, allowing students to work on real-world datasets and build custom AI models.
2. **Industry-Relevant Skills Development** – Students gain expertise in AI model fine-tuning, embedding techniques, and practical applications, making them industry-ready.
3. **Enhancing Creativity and Problem Solving** – The ability to generate human-like content fosters new approaches to solving challenges in media, business automation, and personalized recommendations.
4. **Expanding Career Opportunities** – As AI adoption grows, demand for experts in AI model training, ethical AI development, and prompt engineering increases across domains.
5. **Encouraging AI-Driven Innovation** – Generative AI allows businesses to explore new ideas faster, optimize processes, and build AI-powered creative solutions.

Applications of Generative AI

- **Advanced Chatbots and Conversational AI** – Virtual assistants can respond more naturally and offer human-like interaction.
- **AI in Finance** – Generative AI models are being used for fraud detection, algorithmic trading, and financial forecasting.
- **Code Generation and Software Development** – Tools like GitHub Copilot assist developers by suggesting relevant code snippets and debugging solutions.
- **AI in Marketing and Advertising** – Personalized ad generation, automated social media content creation, and customer sentiment analysis.

Advantages of Learning Generative AI

1. **Ethical AI Considerations** – Understanding bias in AI models and the implications of AI-generated content ensures responsible development and deployment.
2. **Cutting-Edge Research Opportunities** – Generative AI plays a role in groundbreaking research across computational creativity and AI ethics.
3. **AI-powered Automation and Efficiency Gains** – AI-generated content speeds up workflows in content creation, graphic design, and personalized communication.

Course Content Overview

The course delves deeper into:

- **Fine-tuning Pre-trained Models:** Optimizing LLMs for domain-specific tasks.
- **Exploring Transformer Architectures:** Understanding self-attention mechanisms and how they contribute to generative capabilities.
- **Deploying AI Models in Production:** Building scalable AI applications for real-world use cases.
- **Developing Responsible AI:** Addressing bias, fairness, and explainability in generative AI systems

Program 1.

```
In [8]: !pip install gensim
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: gensim in c:\programdata\anaconda3\lib\site-packages
(4.3.0)
Requirement already satisfied: numpy>=1.18.5 in c:\users\student\appdata\roaming\python\python311\site-packages (from gensim) (1.24.4)
Requirement already satisfied: scipy>=1.7.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from gensim) (1.10.1)
Requirement already satisfied: smart-open>=1.8.1 in c:\programdata\anaconda3\lib\site-packages (from gensim) (5.2.1)
Requirement already satisfied: FuzzyTM>=0.4.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from gensim) (2.0.9)
Requirement already satisfied: pandas in c:\users\student\appdata\roaming\python\python311\site-packages (from FuzzyTM>=0.4.0->gensim) (1.5.3)
Requirement already satisfied: pyfume in c:\users\student\appdata\roaming\python\python311\site-packages (from FuzzyTM>=0.4.0->gensim) (0.3.4)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\programdata\anaconda3\lib\site-packages (from pandas->FuzzyTM>=0.4.0->gensim) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\lib\site-packages (from pandas->FuzzyTM>=0.4.0->gensim) (2023.3.post1)
Requirement already satisfied: simpful==2.12.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from pyfume->FuzzyTM>=0.4.0->gensim) (2.12.0)
Requirement already satisfied: fst-psot==1.8.1 in c:\users\student\appdata\roaming\python\python311\site-packages (from pyfume->FuzzyTM>=0.4.0->gensim) (1.8.1)
Requirement already satisfied: minifl in c:\users\student\appdata\roaming\python\python311\site-packages (from fst-psot==1.8.1->pyfume->FuzzyTM>=0.4.0->gensim) (0.0.6)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas->FuzzyTM>=0.4.0->gensim) (1.16.0)
```

Gensim: A Python library for NLP and word embeddings.

50d, 100d, 6B: Dimensions of GloVe vectors and the size of the training corpus.

GloVe embeddings are converted to Word2Vec format for compatibility with libraries like Gensim, which require the Word2Vec format for efficient vector operations and model functionality.

```
In [4]: from gensim.scripts.glove2word2vec import glove2word2vec
from gensim.models import KeyedVectors

# Paths to the GloVe file and output Word2Vec file
glove_input_file = "glove.6B/glove.6B.100d.txt" # Path to GloVe file
word2vec_output_file = "glove.6B/glove.6B.100d.word2vec.txt" # Output file in Word2Vec format

# Convert GloVe format to Word2Vec format
glove2word2vec(glove_input_file, word2vec_output_file)

# Load the converted Word2Vec model
model = KeyedVectors.load_word2vec_format(word2vec_output_file, binary=False)

# Test the Loaded model
print(model.most_similar("hero"))
```

```
C:\Users\student\AppData\Local\Temp\ipykernel_14852\3405237137.py:9: DeprecationWarning: Call to deprecated `glove2word2vec` (KeyedVectors.load_word2vec_format(.., binary=False, no_header=True) loads GLoVE text vectors..).
  glove2word2vec(glove_input_file, word2vec_output_file)
[('heroes', 0.7860509157180786), ('legend', 0.6983181834220886), ('villain', 0.678298830986023), ('man', 0.6675073504447937), ('beloved', 0.6436542272567749), ('heroine', 0.6362980604171753), ('legendary', 0.635901689529419), ('icon', 0.6294612288475037), ('warrior', 0.6287669539451599), ('star', 0.6228683590888977)]
```

Explore Word Relationships

Example 1: Find Similar Words

```
In [5]: similar_to_mysore = model.similar_by_vector(model['bangalore'], topn=3)
print(f"Words similar to 'bangalore': {similar_to_mysore}")
```

```
Words similar to 'bangalore': [('bangalore', 1.0), ('chennai', 0.8824366331100464), ('hyderabad', 0.8300681114196777)]
```

Example 2: Gender Analogy (king - man + woman = queen)

```
In [6]: # Perform vector arithmetic
result_vector_1 = model['waiter'] - model['man'] + model['woman']

# Find the most similar word
result_1 = model.similar_by_vector(result_vector_1, topn=1)
print(f"'waiter - man + woman' = {result_1}")

'waiter - man + woman' = [('queen', 0.8255986571311951)]
```

Example 3: Country-City Relationship (India - Delhi + Bangalore)

```
In [7]: # Perform vector arithmetic
result_vector_2 = model['india'] - model['delhi'] + model['washington']

# Find the most similar word
result_2 = model.similar_by_vector(result_vector_2, topn=3)
print(f'India - Delhi + Washington' = {result_2})

'India - Delhi + Washington' = [('states', 0.8375228643417358), ('united', 0.8281229734420776), ('washington', 0.8155243396759033)]
```

Perform Arithmetic Operations

Example 1: Scaling Vectors

```
In [5]: scaled_vector = model['hotel'] * 2 # Scales the 'king' vector by a factor of 2
result_2 = model.similar_by_vector(scaled_vector, topn=3)

In [6]: result_2

Out[6]: [('hotel', 1.0),
          ('hotels', 0.7933705449104309),
          ('restaurant', 0.7762866020202637)]
```

Example 2: Normalizing Vectors

```
In [11]: import numpy as np
normalized_vector = model['fish'] / np.linalg.norm(model['fish'])
result_2 = model.similar_by_vector(normalized_vector, topn=3)

In [12]: result_2

Out[12]: [('fish', 1.0), ('shrimp', 0.7793381810188293), ('salmon', 0.760814368724823)]
```

Example 3: Averaging Vectors

```
In [13]: average_vector = (model['king'] + model['woman'] + model['man']) / 3
result_2 = model.similar_by_vector(average_vector, topn=3)
result_2

Out[13]: [('man', 0.9197071194648743),
          ('woman', 0.8637868165969849),
          ('father', 0.8270207047462463)]
```

Model Comparision

```
In [14]: # Paths to the GLoVe file and output Word2Vec file
glove_input_file = "glove.6B/glove.6B.50d.txt" # Path to GLoVe file
word2vec_output_file = "glove.6B/glove.6B.50d.word2vec.txt" # Output file in Word2Vec

# Convert GLoVe format to Word2Vec format
glove2word2vec(glove_input_file, word2vec_output_file)

# Load the converted Word2Vec model
model_50d = KeyedVectors.load_word2vec_format(word2vec_output_file, binary=False)

C:\Users\student\AppData\Local\Temp\ipykernel_13768\1425925041.py:6: DeprecationWarning: Call to deprecated `glove2word2vec` (KeyedVectors.load_word2vec_format(.., binary=False, no_header=True) loads GLoVE text vectors.).
    glove2word2vec(glove_input_file, word2vec_output_file)
```



```
In [17]: # Paths to the GLoVe file and output Word2Vec file
glove_input_file = "glove.6B/glove.6B.100d.txt" # Path to GLoVe file
word2vec_output_file = "glove.6B/glove.6B.100d.word2vec.txt" # Output file in Word2Vec

# Convert GLoVe format to Word2Vec format
glove2word2vec(glove_input_file, word2vec_output_file)

# Load the converted Word2Vec model
model_100d = KeyedVectors.load_word2vec_format(word2vec_output_file, binary=False)

C:\Users\student\AppData\Local\Temp\ipykernel_13768\218754363.py:6: DeprecationWarning: Call to deprecated `glove2word2vec` (KeyedVectors.load_word2vec_format(.., binary=False, no_header=True) loads GLoVE text vectors.).
    glove2word2vec(glove_input_file, word2vec_output_file)
```

Similarity: Measures how close two vectors are in direction using cosine similarity. A value closer to 1 indicates high similarity, while 0 means orthogonal and -1 means opposite directions.

Distance: Measures how far two vectors are from each other in space (e.g., Euclidean distance). A smaller distance indicates the vectors are more similar in magnitude and direction.

Calculate similarity between two words

```
In [18]: word1 = "hospital"
word2 = "doctor"

# Similarity in 50d
similarity_50d = model_50d.similarity(word1, word2)

# Similarity in 100d
similarity_100d = model_100d.similarity(word1, word2)

# Results
print(f"Similarity (50d) between '{word1}' and '{word2}': {similarity_50d:.4f}")
print(f"Similarity (100d) between '{word1}' and '{word2}': {similarity_100d:.4f}")

Similarity (50d) between 'hospital' and 'doctor': 0.6724
Similarity (100d) between 'hospital' and 'doctor': 0.6901
```

Calculate distance between two words

```
In [18]: # Calculate distance between two words
distance_50d = model_50d.distance(word1, word2)
distance_100d = model_100d.distance(word1, word2)

# Results
print(f"Distance (50d) between '{word1}' and '{word2}': {distance_50d:.4f}")
print(f"Distance (100d) between '{word1}' and '{word2}': {distance_100d:.4f}")

Distance (50d) between 'hospital' and 'doctor': 0.3276
Distance (100d) between 'hospital' and 'doctor': 0.3099
```

Analysis of Results

1. 'actor - man + woman' = actress (0.916)

- The result confirms that the model has captured gender analogies, where subtracting "man" and adding "woman" to "actor" produces the semantically related word "actress."

2. 'India - Delhi + Washington' = ['states', 0.838], ['united', 0.828], ['washington', 0.816]

- The arithmetic operation shows that "India - Delhi + Washington" produces words like "states" and "united," suggesting a shift from a city to broader political entities, such as countries or states.

3. Scaling Vectors ('hotel' * 2) = [('hotel', 1.0), ('hotels', 0.793), ('restaurant', 0.776)]

- The scaled vector results in "hotel" being the most similar to itself, and its plural form "hotels" is the second most similar, followed by related terms like "restaurant."

4. Normalizing Vectors ('fish') = [('fish', 1.0), ('shrimp', 0.779), ('salmon', 0.761)]

- Normalizing the vector for "fish" leads to very similar words like "shrimp" and "salmon," which are semantically related types of fish.

5. Averaging Vectors ('king' + 'woman' + 'man') / 3 = [('man', 0.920), ('woman', 0.864), ('father', 0.827)]

- Averaging the vectors of "king," "woman," and "man" results in "man" and "woman" being the most similar words, indicating that the averaged vector represents a central concept of human relationships.

6. Similarity and Distance Calculation for 'hospital' and 'doctor':

- **Similarity:** 0.6724 (50d) vs. 0.6901 (100d)
 - The similarity between "hospital" and "doctor" is higher in the 100d model, indicating that the higher-dimensional model captures the relationship between these words more accurately.
- **Distance:** 0.3276 (50d) vs. 0.3099 (100d)
 - The distance between "hospital" and "doctor" is smaller in the 100d model, confirming that the 100d model finds them closer in the vector space, aligning with the similarity results.

Conclusion

- Higher-dimensional models (100d) generally provide more accurate and nuanced word relationships, both in terms of **similarity** and **distance**.
- Arithmetic operations like scaling, averaging, and vector shifts (analogies) allow deeper exploration of word meanings and relationships, and these can vary slightly with model dimensions.

Program 2.

Use dimensionality reduction (e.g., PCA or t-SNE) to visualize word embeddings for Q 1. Select 10 words from a specific domain (e.g., sports, technology) and visualize their embeddings. Analyze clusters and relationships. Generate contextually rich outputs using embeddings. Write a program to generate 5 semantically similar words for a given input.

```
In [2]: import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from gensim.models import KeyedVectors

# Load pre-trained GloVe embeddings (100d model)
model_100d = KeyedVectors.load_word2vec_format("glove.6B/glove.6B.100d.word2vec.txt", binary=False)

# Select 10 words from a specific domain (sports) # Included other words to show how embeddings are different
words = ['football', 'soccer', 'basketball', 'tennis', 'engineer', 'information', 'baseball', 'coach', 'goal', 'player',
         'word_vectors = np.array([model_100d[word] for word in words])

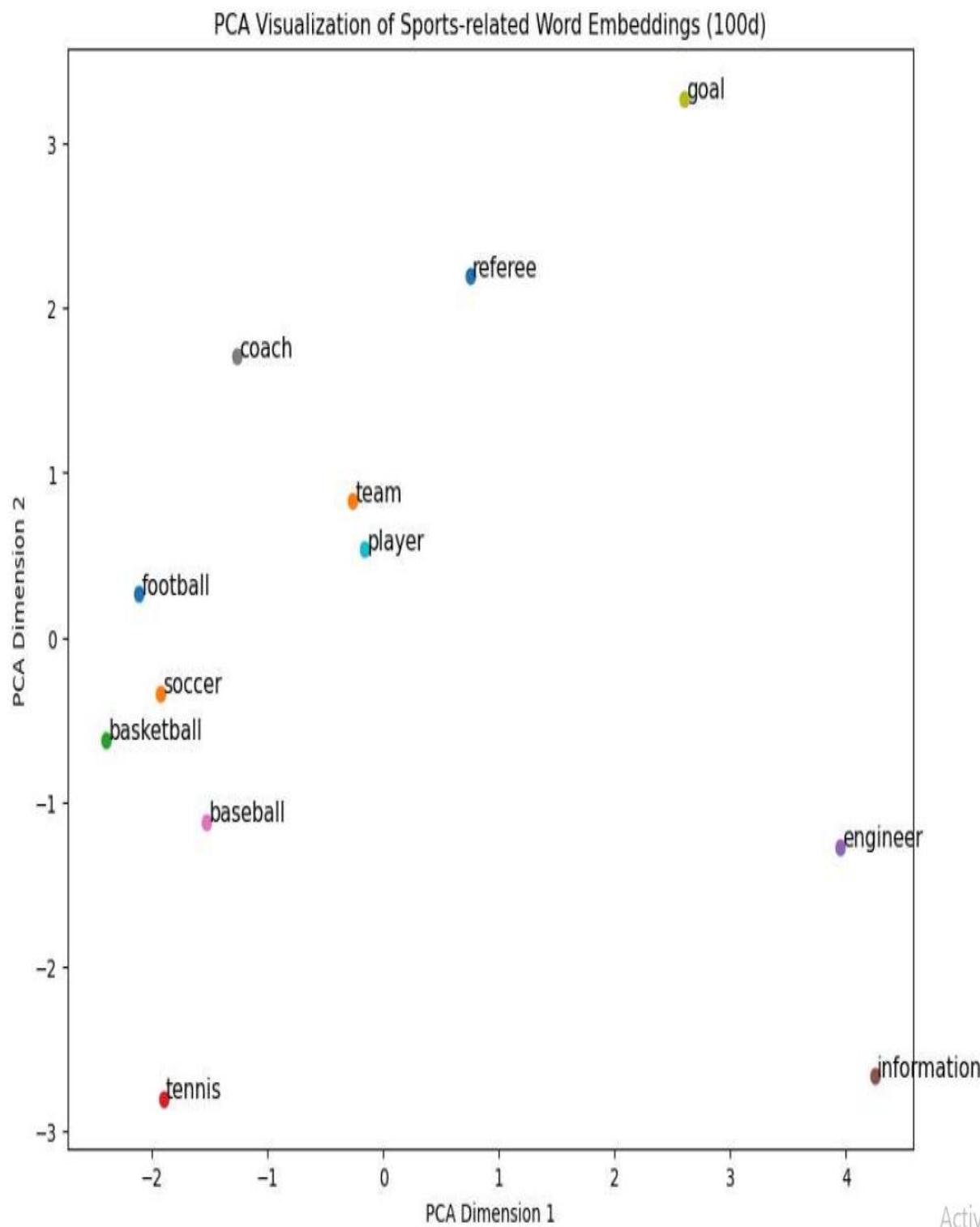
# Dimensionality reduction using PCA
# Using PCA to reduce to 2D for visualization
pca = PCA(n_components=2)
pca_result = pca.fit_transform(word_vectors)

# Plotting the words in 2D space
plt.figure(figsize=(10, 8))
for i, word in enumerate(words):
    plt.scatter(pca_result[i, 0], pca_result[i, 1])
    plt.text(pca_result[i, 0] + 0.02, pca_result[i, 1], word, fontsize=12)
plt.title("PCA Visualization of Sports-related Word Embeddings (100d)")
plt.xlabel("PCA Dimension 1")
plt.ylabel("PCA Dimension 2")
plt.show()

# 5 Semantically Similar Words Generator Function
def get_similar_words(word, model, topn=5):
    similar_words = model.similar_by_word(word, topn=topn)
    return similar_words
```

Activati
Go to Set

```
# Example: Get 5 words similar to "football"
similar_words_football = get_similar_words('football', model_100d, topn=5)
print(f"Words similar to 'football': {similar_words_football}")
```



Vector embedding for 'football':

```
[ 0.43865  0.10537  0.45972 -1.0724  -1.2471   0.76351
  0.47528  0.083857 -0.9127  -0.27328 -0.018591  -1.184
  0.22748  0.16847  -0.52158  0.11339  1.3757   0.11892
 -0.37683  0.51149  -0.8833  0.96259  0.18143  -0.407
  0.036181 -0.74432 -0.0027401 -0.70068  0.53103  0.45114
 -0.72884  1.0631   -0.28008  -0.63848  0.15645  -0.46927
 -1.0071   1.033    -1.4354  -0.27485  0.048984  0.13951
  0.43072  -0.78791  0.41097  0.58509  1.0155  -0.1839
  0.27487  -0.90866 -0.30441  -0.17396  0.020941  0.62813
  0.10978  -2.3885  -0.56364  -0.27193  0.98728  0.70608
 -0.512    0.52636  -0.78503  -0.68714  0.38121  0.097582
 -0.20237  0.43208  -0.30527  0.57925  0.62619  -0.47415
  0.33834  -0.28421 -0.097465  0.19597  0.54849  0.59918
 -0.41576  0.1021   0.6766   0.0042009 -0.12354  -0.76613
 -0.27436  -0.68248 -1.0789  -0.16708  0.81671  0.026999
 -0.38707  0.40448  -1.0995  0.64718  -0.12802  -0.26084
 -0.96701  0.88078  1.012   -0.022223 ]
```

Vector embedding for 'soccer':

```
[ 8.3777e-01  5.1890e-01  6.4015e-01 -6.2606e-01 -9.7474e-01  1.0127e+00
 6.2729e-02  4.4316e-01 -8.3299e-01  7.9888e-02 -1.1815e-02 -1.1265e+00
 1.2554e-01 -3.4206e-01 -5.1422e-01  3.8526e-01  1.0032e+00 -1.5172e-03
 -2.2684e-01  3.5658e-01 -6.2449e-01  8.7271e-01  3.6670e-01  4.6462e-01
 -1.0046e-01 -4.4798e-01 -2.1813e-01 -5.6423e-01  5.6665e-01  5.1601e-01
 -5.6511e-01  7.1919e-01 -6.5347e-01 -9.5952e-02  5.6028e-01 -4.9956e-01
 -7.4757e-01  6.8516e-01 -1.4518e+00 -1.1207e-01  1.0241e-01  3.0537e-02
 1.1326e-02 -8.6873e-01  6.3622e-01  4.9539e-01  3.0538e-01  7.7133e-02
 7.4048e-02 -7.1163e-01 -1.9159e-01 -3.4168e-01 -4.7185e-01  5.6794e-01
 3.7454e-01 -1.9207e+00 -8.6040e-01  5.7058e-01  1.0700e+00  9.2101e-01
 -6.4825e-01  5.3516e-01 -1.5556e-01 -9.0021e-01 -1.7459e-01  3.3146e-02
 -5.7512e-01  2.9963e-01 -4.0008e-01 -1.0765e-01  4.1384e-01 -7.2178e-01
 1.1442e-01 -2.1291e-01  5.4949e-02  1.3213e-01  7.8766e-01  8.9291e-02
 -6.6689e-01  3.3998e-01  9.7163e-01 -8.4871e-02  1.7542e-01 -4.6039e-01
 -8.5885e-02 -7.5960e-01 -1.5071e+00  2.1545e-01  2.1209e-01 -4.4837e-01
 -2.5882e-01  3.3814e-01 -4.7979e-01  2.1059e-01  2.3621e-01 -3.6699e-01
 -8.1440e-01  5.4515e-01  9.7946e-01  2.3367e-01]
```

Program 3

Train a custom Word2Vec model on a small dataset. Train embeddings on a domain-specific corpus (e.g., legal, medical) and analyze how embeddings capture domain-specific semantics.

Important Steps

1. **Tokenization:** Converts sentences into lists of lowercase tokens for processing.
2. **Word2Vec Training:**
 - `vector_size` : Sets the embedding dimension to 50.
 - `window` : Uses a context window of 3 words.
 - `sg` : Skip-gram (`sg=1`) is used, which works better for smaller datasets.
 - `epochs` : The number of training iterations.
3. **Visualization:** PCA reduces the high-dimensional word vectors to 2D for visualization, helping to understand semantic relationships.
4. **Semantic Analysis:** The `most_similar` method identifies words that are semantically similar based on embeddings.

Example: Legal Corpus

```
In [3]: from gensim.models import Word2Vec
from gensim.utils import simple_preprocess
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
```

```
In [4]: legal_corpus = [
    "The court ruled in favor of the plaintiff.",
    "The defendant was found guilty of negligence.",
    "A breach of contract case was filed.",
    "The agreement between parties must be honored.",
    "The lawyer presented compelling evidence.",
    "Legal documents must be drafted carefully.",
    "The jury deliberated for several hours.",
    "A settlement was reached between the parties.",
    "The plaintiff claimed damages for losses incurred.",
    "The contract outlined the obligations of both parties."
]
```

Activate
Go to Sett

```
In [5]: # Example legal corpus
legal_corpus = [
    "The court ruled in favor of the plaintiff.",
    "The defendant was found guilty of negligence.",
    "A breach of contract case was filed.",
    "The agreement between parties must be honored.",
    "The lawyer presented compelling evidence.",
    "Legal documents must be drafted carefully.",
    "The jury deliberated for several hours.",
    "A settlement was reached between the parties.",
    "The plaintiff claimed damages for losses incurred.",
    "The contract outlined the obligations of both parties."
]

# Preprocess the corpus
tokenized_corpus = [simple_preprocess(sentence) for sentence in legal_corpus]

# Train the Word2Vec model
legal_word2vec = Word2Vec(
    sentences=tokenized_corpus,
    vector_size=50, # Embedding dimension
    window=3,       # Context window size
    min_count=1,   # Minimum word frequency
    sg=1,          # Skip-gram model
    epochs=100      # Training epochs
)

# Save the model for later use
legal_word2vec.save("legal_word2vec.model")
```

```
In [6]: # Train the Word2Vec model
legal_word2vec = Word2Vec(
    sentences=tokenized_corpus,
    vector_size=50, # Embedding dimension
    window=3,       # Context window size
    min_count=1,   # Minimum word frequency
    sg=1,          # Skip-gram model
    epochs=100      # Training epochs
)

# Save the model for later use
legal_word2vec.save("legal_word2vec.model")
```

Activat

```
In [7]: # Analyze embeddings: Display vector for a specific word
word = "lawyer"
if word in legal_word2vec.wv:
    print(f"Vector embedding for '{word}':\n{legal_word2vec.wv[word]}\n")
else:
    print(f"Word '{word}' not found in the Word2Vec model.")

Vector embedding for 'lawyer':
[ 0.00373483  0.01353383  0.00585796 -0.01324683  0.01500349 -0.01261986
  0.01892563  0.00698961 -0.0087639 -0.01023367 -0.00875896 -0.01318524
  0.01972703 -0.00463062  0.01525868 -0.01837575  0.0055629 -0.00126356
  0.01417167 -0.01969541  0.01564029 -0.00948072 -0.0107858 -0.01128642
 -0.00610619 -0.00604345 -0.00693252 -0.01396556  0.00086967 -0.00136903
 -0.00358557  0.00685404 -0.01432065 -0.00657563  0.00952303  0.01720192
 -0.01858611  0.01418636  0.01038651 -0.00818817  0.01832661 -0.01858529
  0.01404059  0.01154918  0.00326395 -0.01036671 -0.00841038 -0.00736812
  0.00374052  0.00413726]
```

```
In [8]: # Visualize embeddings using PCA
words_to_visualize = ["court", "plaintiff", "defendant", "agreement", "lawyer", "evidence", "contract", "settlement", ""]
word_vectors = [legal_word2vec.wv[word] for word in words_to_visualize]
```

```
In [9]: word_vectors
```

```
Out[9]: [array([-0.01018794, -0.0037532 , -0.01479373,  0.00535417,  0.00549183,
 -0.00194653, -0.00904275, -0.00120178,  0.01239534,  0.005502 ,
 -0.01752885, -0.00888894,  0.00678894,  0.00598825, -0.01972261,
  0.01158325, -0.01438892, -0.01200779,  0.00463451, -0.01056976,
  0.00906795,  0.01991566, -0.00384839,  0.01845003,  0.00452612,
  0.02153785,  0.0106156 , -0.0164802 , -0.0075984 ,  0.01259563,
  0.01069134,  0.01610584,  0.01608272,  0.01619358, -0.02157517,
  0.00898223, -0.00762749,  0.00642556,  0.01106042,  0.00757853,
  0.01795846,  0.00227335, -0.00347768,  0.01356644, -0.00962057,
  0.00016249,  0.01841913, -0.01246461,  0.00897428, -0.01424266],
 dtype=float32),
 array([-0.01382369,  0.0011437 , -0.01449836, -0.00296123,  0.00624782,
  0.00982854,  0.00482432,  0.00674102, -0.01035763,  0.0125059 ,
 -0.01251031,  0.00691753, -0.01420641,  0.00583739, -0.01051113,
 -0.00608784, -0.00284186,  0.01448168,  0.00904517, -0.01370935,
  0.00321437, -0.01510567,  0.01918532,  0.01829434, -0.00426899,
  0.00343321,  0.00058916,  0.01309333, -0.0183534 ,  0.00069488,
  0.0132396 ,  0.0028656 ,  0.00451153, -0.01875341,  0.01468391,
 -0.01201477, -0.00313035,  0.00620906, -0.0025351 ,  0.00151398,
  0.0066815 , -0.01435055, -0.02045849,  0.01987134,  0.01433266,
 -0.01331776,  0.00661788, -0.00128313,  0.01081608, -0.01213262],
 dtype=float32),
 array([-0.01735372,  0.00324048, -0.00153466, -0.01745638, -0.01992291,
 -0.00444436,  0.01032289,  0.00879421, -0.01455397, -0.01536747,
 -0.01001598, -0.00653257, -0.0128883 , -0.01829896, -0.0059358 ,
 -0.01495283, -0.00974466, -0.00899372, -0.00697665, -0.00573702,
 -0.01700949,  0.0003172 ,  0.01874463,  0.01480774, -0.01384414,
 -0.00612229,  0.00565069, -0.01732042,  0.00195022,  0.01274919,
  0.01080692, -0.01920946, -0.00795812, -0.01638088, -0.00148547,
  0.01870203,  0.01399906,  0.00958245,  0.00941055, -0.00658938,
  0.02153141, -0.01520018, -0.01545056, -0.00327682,  0.00024155,
 -0.00606883, -0.00135896,  0.01406399,  0.00023136, -0.00163963],
 dtype=float32),
 array([ 0.00544287,  0.01555425, -0.00307019,  0.01717134,  0.00693973,
 -0.0170452 , -0.00679161, -0.00333116,  0.00903156, -0.00295565,
 -0.00578579,  0.01436892,  0.02001157, -0.00213262, -0.01079166,
 -0.007799 , -0.00751752, -0.01736892,  0.00095211, -0.01050753,
  0.00615652,  0.01262798, -0.00638232, -0.01966911,  0.00394809,
 -0.01237557,  0.0045896 , -0.00610946,  0.01346509,  0.00106505,
  0.00631289, -0.00667795, -0.00218376,  0.01535427,  0.00144457,
 -0.0117866 , -0.01405202,  0.00186158,  0.0125593 ,  0.0105592 ,
 -0.01650265,  0.01693893,  0.0074757 ,  0.0163192 ,  0.02056517,
 -0.01457632, -0.01834234,  0.01092377,  0.01994778,  0.00864314],
 dtype=float32),
 array([ 0.00373483,  0.01353383,  0.00585796, -0.01324683,  0.01500349,
```

```

-0.01261986,  0.01892563,  0.00698961, -0.0087639 , -0.01023367,
-0.00875896, -0.01318524,  0.01972703, -0.00463062,  0.01525868,
-0.01837575,  0.0055629 , -0.00126356,  0.01417167, -0.01969541,
0.01564029, -0.00948072, -0.0107858 , -0.01128642, -0.00610619,
-0.00604345, -0.00693252, -0.01396556,  0.00086967, -0.00136903,
-0.00358557,  0.00685404, -0.01432065, -0.00657563,  0.00952303,
0.01720192, -0.01858611,  0.01418636,  0.01038651, -0.00818817,
0.01832661, -0.01858529,  0.01404059,  0.01154918,  0.00326395,
-0.01036671, -0.00841038, -0.00736812,  0.00374052,  0.00413726],
dtype=float32),
array([ 0.00550566,  0.00103798, -0.00515228,  0.01945088,  0.00499871,
0.00736707, -0.0011947 ,  0.00298867,  0.01247635, -0.00248031,
0.00660107, -0.00238972,  0.01178223,  0.00798718,  0.00505932,
-0.00936528, -0.00755702,  0.00989482, -0.01304692, -0.00193519,
-0.00039899,  0.0078729 , -0.01549838,  0.01741308, -0.0023178 ,
-0.00983727,  0.00754468, -0.0027872 , -0.01603217, -0.00921708,
-0.00134961, -0.01871502,  0.002125 ,  0.00480915, -0.00744796,
0.00537565,  0.00629158,  0.01973929,  0.0024904 ,  0.00340102,
0.00710946, -0.00441335, -0.01761757,  0.01698 , -0.0031966 ,
-0.0194808 , -0.01307702, -0.00849545,  0.00867249,  0.01145031],
dtype=float32),
array([ 0.00298495, -0.00700375, -0.01431873, -0.01400242, -0.01991 ,
-0.01428693, -0.00105788, -0.00551727, -0.0153189 , -0.0100668 ,
0.00858984, -0.01069687,  0.01958971,  0.00508815, -0.01531299,
0.02237322,  0.01962719,  0.01488377, -0.01710452,  0.00707861,
0.01021231,  0.01304598,  0.01277774,  0.00337116, -0.00486931,
0.01909359,  0.01800028,  0.01032766, -0.00758116, -0.00048564,
0.00164387, -0.018941 , -0.01410591, -0.00047871, -0.00010738,
-0.01102702, -0.0061726 , -0.01550268,  0.0161471 , -0.00069464,
-0.00545253,  0.01093123,  0.01718066, -0.00617879,  0.0186232 ,
0.01143978,  0.01163601, -0.00062903,  0.01886317, -0.0114121 ],
dtype=float32),
array([-0.00976338, -0.00780081,  0.019559 ,  0.01830878, -0.00654693,
0.01011876,  0.01784409, -0.00301464,  0.01761319,  0.01262996,
0.00393919, -0.00980376, -0.00886089, -0.00519702,  0.01557352,
-0.01077065, -0.00356288,  0.02101176,  0.00479641,  0.01005143,
-0.01529913,  0.00012613,  0.01357427, -0.01018804,  0.01573833,
0.02000298, -0.00148577, -0.00239202, -0.00189635, -0.01172749,
-0.01742925, -0.00479667, -0.00404787,  0.00869905, -0.01522061,
0.01094797, -0.01160657, -0.0163735 ,  0.01649981,  0.01770434,
-0.00497504,  0.00637913, -0.01261914, -0.0161758 , -0.00964475,
0.01381735,  0.01255536,  0.01808335,  0.01568656,  0.01504712],
dtype=float32),
array([ 0.01492715,  0.01934095,  0.01774599, -0.00747902,  0.01891196,
-0.0020531 ,  0.01053821,  0.00635226, -0.00197045,  0.00632444,

```

```

-0.01059867, -0.01259004, -0.01428318,  0.00465197,  0.01267453,
 0.0028981 ,  0.00384051,  0.00779968,  0.01519439, -0.01727828,
 0.00548228, -0.01392599,  0.00899558,  0.01923842,  0.01556924,
 0.01372755,  0.01566461,  0.01412691,  0.01313736,  0.01728814,
 -0.01028677,  0.01760968,  0.01114872, -0.00440745,  0.01607866,
 0.01023387,  0.02058647,  0.00533376,  0.01917837,  0.00176341,
 0.01960336,  0.0070012 ,  0.01266869, -0.00624314,  0.01447076,
 0.01456392, -0.00432669, -0.00459186,  0.00780778, -0.01304201],
dtype=float32),
array([ 0.0158812 ,  0.0174168 ,  0.00195527, -0.01554414,  0.01595952,
 -0.00898288,  0.0134456 ,  0.01096715,  0.01782416, -0.02043355,
 0.01853634, -0.02043897, -0.01187125, -0.01672532, -0.01152777,
 0.01697107,  0.02129747, -0.00410723,  0.0053023 , -0.01103053,
 0.017639 ,  0.01337754,  0.0028419 , -0.00731513, -0.01343816,
 0.01128781,  0.00173393, -0.00338352,  0.01469343, -0.00834176,
 -0.01866035, -0.00566033,  0.00234445, -0.0135692 , -0.0126051 ,
 -0.01704373,  0.02071049,  0.0147259 , -0.00145971,  0.01323994,
 0.01840546, -0.0020906 ,  0.0126531 ,  0.0093615 ,  0.01196339,
 0.01458218, -0.00961088, -0.00608193,  0.00305689,  0.01033708],
dtype=float32)]

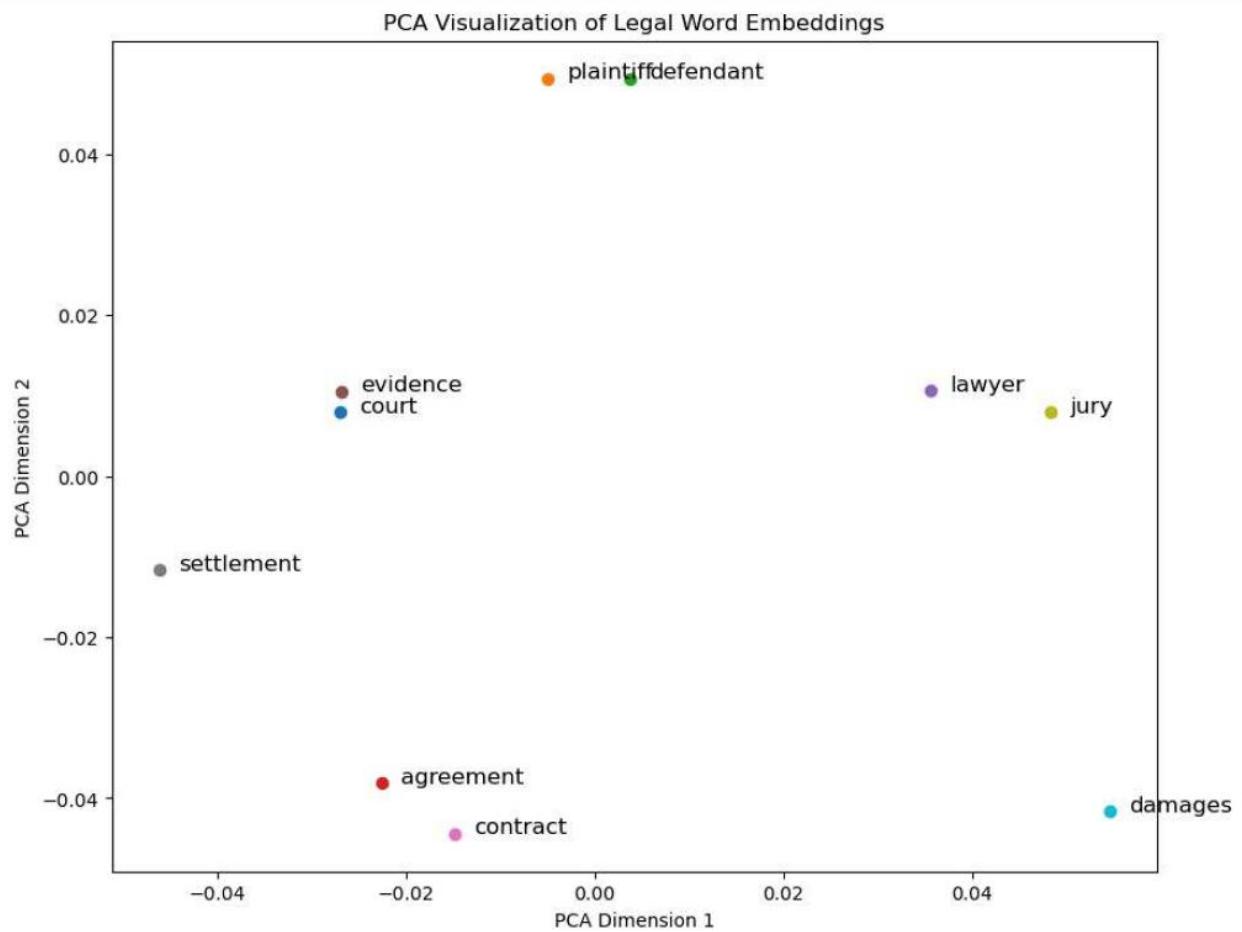
```

```
In [10]: # Dimensionality reduction
pca = PCA(n_components=2)
reduced_vectors = pca.fit_transform(word_vectors)
reduced_vectors
```

```
Out[10]: array([[-0.02688162,  0.00792018],
 [-0.00493226,  0.04934309],
 [ 0.00377306,  0.04936944],
 [-0.02256997, -0.03808062],
 [ 0.0355795 ,  0.01066101],
 [-0.02682294,  0.01050709],
 [-0.01486912, -0.0443972 ],
 [-0.04605154, -0.01166099],
 [ 0.0482769 ,  0.0079725 ],
 [ 0.05449799, -0.0416345 ]])
```

```
In [70]: # Plot embeddings
plt.figure(figsize=(10, 8))
for i, word in enumerate(words_to_visualize):
    plt.scatter(reduced_vectors[i, 0], reduced_vectors[i, 1])
    plt.text(reduced_vectors[i, 0] + 0.002, reduced_vectors[i, 1], word, fontsize=12)
plt.title("PCA Visualization of Legal Word Embeddings")
plt.xlabel("PCA Dimension 1")
```

```
plt.ylabel("PCA Dimension 2")
plt.show()
```



```
In [12]: # Find similar words
similar_words = legal_word2vec.wv.most_similar("lawyer", topn=8)
print(f"Words similar to 'lawyer': {similar_words}")

Words similar to 'lawyer': [('carefully', 0.29186686873435974), ('claimed', 0.27888569235801697), ('jury', 0.21892617642879486), ('damages', 0.1961500644683838), ('negligence', 0.1820133775472641), ('documents', 0.15884239971637726), ('both', 0.15777595341205597), ('defendant', 0.15757431089878082)]
```

Example: Legal and Medical / Healthcare Corpus

```
In [2]: from gensim.models import Word2Vec
from gensim.utils import simple_preprocess
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Enhanced Legal and medical corpus
enhanced_corpus = [
    # Legal domain
    "The court ordered the immediate release of the detained individual due to lack of evidence.",
    "A new amendment was introduced to ensure the protection of intellectual property rights.",
    "The defendant pleaded not guilty, citing an alibi supported by credible witnesses.",
    "The plaintiff accused the company of violating environmental regulations.",
    "A settlement agreement was reached through arbitration, avoiding a lengthy trial.",
    "The legal team presented a compelling argument to overturn the previous judgment.",
    "Contractual obligations must be fulfilled unless waived by mutual consent.",
    "The jury found the accused guilty of fraud and embezzlement.",
    "The appeal was dismissed as the evidence presented was deemed inadmissible.",
    "The attorney emphasized the importance of adhering to constitutional rights.",

    # Medical domain
    "The patient was admitted to the emergency department with severe chest pain.",
    "The surgeon successfully performed a minimally invasive procedure to remove the tumor.",
    "Clinical trials showed significant improvement in patients treated with the experimental drug.",
    "Regular screening is essential for early detection of chronic illnesses such as diabetes.",
    "The doctor recommended physical therapy to improve mobility after surgery.",
    "The hospital implemented stringent protocols to prevent the spread of infectious diseases.",
    "The nurse monitored the patient's vital signs hourly to ensure stability.",
    "Vaccination campaigns have drastically reduced the prevalence of polio worldwide.",
    "The radiologist identified a small abnormality in the CT scan requiring further investigation.",
    "Proper nutrition and exercise are vital components of a healthy lifestyle."
]
```

Annotate

```
In [3]: # Preprocess the corpus
tokenized_corpus = [simple_preprocess(sentence) for sentence in enhanced_corpus]
tokenized_corpus
```

```
Out[3]: [[['the',
  'court',
  'ordered',
  'the',
  'immediate',
  'release',
  'of',
  'the',
  'detained',
  'individual',
  'due',
  'to',
  'lack',
  'of',
  'evidence'],
  ['new',
  'amendment',
  'was',
  'introduced',
  'to',
  'ensure',
  'the',
  'protection',
  'of',
  'intellectual',
  'property',
  'rights'],
  ['the',
  'defendant',
  'pleaded',
  'not',
  'guilty',
  'citing',
  'an',
  'alibi',
  'supported',
  'by',
  'credible',
  'witnesses'],
  ['the',
  'plaintiff',
  'accused',
  'the',
  'company',
  'of'],
```

```
'violating',
'environmental',
'regulations'],
['settlement',
'agreement',
'was',
'reached',
'through',
'arbitration',
'avoiding',
'lengthy',
'trial'],
['the',
'legal',
'team',
'presented',
'compelling',
'argument',
'to',
'overtur',
'the',
'previous',
'judgment'],
['contractual',
'obligations',
'must',
'be',
'fulfilled',
'unless',
'waived',
'by',
'mutual',
'consent'],
['the',
'jury',
'found',
'the',
'accused',
'guilty',
'of',
'fraud',
'and',
'embezzlement'],
['the',
'appeal',
```

```
'was',
'dismissed',
'as',
'the',
'evidence',
'presented',
'was',
'deemed',
'inadmissible'],
['the',
'attorney',
'emphasized',
'the',
'importance',
'of',
'adhering',
'to',
'constitutional',
'rights'],
['the',
'patient',
'was',
'admitted',
'to',
'the',
'emergency',
'department',
'with',
'severe',
'chest',
'pain'],
['the',
'surgeon',
'successfully',
'performed',
'minimally',
'invasive',
'procedure',
'to',
'remove',
'the',
'tumor'],
['clinical',
'trials',
'showed',
```

```
'significant',
'improvement',
'in',
'patients',
'treated',
'with',
'the',
'experimental',
'drug'],
['regular',
'screening',
'is',
'essential',
'for',
'early',
'detection',
'of',
'chronic',
'illnesses',
'such',
'as',
'diabetes'],
['the',
'doctor',
'recommended',
'physical',
'therapy',
'to',
'improve',
'mobility',
'after',
'surgery'],
['the',
'hospital',
'implemented',
'strict',
'protocols',
'to',
'prevent',
'the',
'spread',
'of',
'infectious',
'diseases'],
['the',
```

```
'nurse',
'monitored',
'the',
'patient',
'vertical',
'signs',
'hourly',
'to',
'ensure',
'stability'],
['vaccination',
'campaigns',
'have',
'dramatically',
'reduced',
'the',
'prevalence',
'of',
'polio',
'worldwide'],
['the',
'radiologist',
'identified',
'small',
'abnormality',
'in',
'the',
'ct',
'scan',
'requiring',
'further',
'investigation'],
['proper',
'nutrition',
'and',
'exercise',
'are',
'vertical',
'components',
'of',
'healthy',
'lifestyle']]
```

```
In [4]: # Train Word2Vec
domain_word2vec = Word2Vec(
    sentences=tokenized_corpus,
    vector_size=100, # Higher embedding dimension for better representation
    window=5,        # Wider context window
    min_count=1,     # Include all words
    sg=1,            # Skip-gram model
    epochs=150       # More training iterations
)

In [5]: # Save the model
domain_word2vec.save("enhanced_domain_word2vec.model")

In [17]: # Analyze embeddings: Get vectors for specific words
words_to_analyze = ["court", "plaintiff", "doctor", "patient", "guilty", "surgery"]
for word in words_to_analyze:
    if word in domain_word2vec.wv:
        print(f"Vector embedding for '{word}':\n{domain_word2vec.wv[word]}\n")
    else:
        print(f"Word '{word}' not found in the Word2Vec model.")
```

Vector embedding for 'court':

```
[ 0.00017663  0.00743831  0.00969632  0.01124777 -0.00897442  0.00472662
 -0.00234596  0.00374601 -0.00157454  0.00307687  0.00721988  0.00014851
  0.00989491  0.00376527 -0.00713342 -0.00904652  0.01237829 -0.00458163
 -0.00523104 -0.00367172  0.00086233 -0.005187  0.00466362 -0.00047746
  0.00255735  0.00753297 -0.00821684  0.00420135  0.00488515  0.00365502
  0.00959129 -0.00758941  0.01119815 -0.0149827  0.00201963 -0.00759506
 -0.00190555 -0.00895344 -0.00489069 -0.00779137 -0.0048753  0.00021034
  0.00113659 -0.00123351 -0.00665795 -0.0110082 -0.00152741 -0.00707312
 -0.00692021  0.00271541  0.00726788  0.00474315 -0.01170204  0.00251061
 -0.00676082 -0.01016952  0.00286596 -0.00155355 -0.0092033 -0.00646701
 -0.00384859  0.00358914 -0.00190337  0.01002331 -0.00630275  0.00711972
  0.007339   0.00822473 -0.01003852  0.00638923  0.00622656  0.0100633
  0.0042153  -0.00290617  0.00460781 -0.00649033  0.00379658  0.00789798
  0.00182467  0.00832622 -0.01469911 -0.00698977  0.00589895  0.00583221
 -0.00793246 -0.00124807  0.00828708 -0.00274233 -0.00459441  0.00733121
  0.01148863 -0.00683292  0.00192717 -0.00482216 -0.00112542  0.0010045
 -0.00459567  0.00734376 -0.0069657  0.00531272]
```

Vector embedding for 'plaintiff':

```
[-1.0662489e-02  9.7385598e-03  3.4756137e-03  1.1294241e-02
 5.8995499e-03  1.3437573e-03  1.3426265e-02  2.2543095e-04
 -9.2070960e-03  9.0760761e-04  6.3291392e-03 -1.0475398e-02
 1.0081153e-02 -8.5829583e-05  8.2821092e-03 -2.3233807e-03
 1.5971650e-04 -1.7414209e-03 -9.7222058e-03 -1.1336627e-02
 4.3367324e-03  1.0453109e-02  1.7080482e-02 -8.7737506e-03
 3.7531380e-03  3.8113615e-03 -5.9637344e-03 -8.0534685e-03
 -1.4423394e-03 -6.4051617e-03  1.1007547e-02 -8.9003462e-03
 1.0641597e-02  1.7153923e-03 -1.7465688e-03  6.2488914e-03
 -1.3276789e-03  7.9928357e-03  5.2064378e-04 -1.1779488e-02
 -3.6209833e-03  2.3854021e-03 -1.7902178e-03 -8.9073041e-04
 5.3480541e-04 -7.4649383e-03 -2.0129688e-03  8.1461715e-04
 -1.5493475e-03 -1.1691412e-04  4.6856981e-04 -1.0649770e-03
 -1.6116065e-03 -5.3310040e-03  3.4014264e-03 -3.3061029e-03
 4.9278443e-03 -5.3212477e-04  3.2568546e-03 -7.1880710e-03
 -8.2602138e-03 -8.8413237e-03  1.5619625e-02  1.3455788e-03
 -1.5836736e-02  5.9275334e-03 -6.8035880e-03  9.8182093e-03
 -1.6868895e-02  5.8864909e-03 -6.5259803e-03  5.2828933e-03
 1.0591622e-02  8.8061299e-03 -4.3318677e-03 -7.4615045e-03
 -4.5252265e-03 -4.5578824e-03 -1.3106052e-02 -3.0325642e-03
 -8.5064340e-03 -9.2060845e-03  6.0112868e-03  5.9379553e-03
 4.6743322e-03 -9.7127270e-04  6.5881531e-03 -7.1246484e-03
 -1.6362706e-03  6.8450025e-03 -3.0371335e-03  1.6706325e-03
 5.0919107e-03 -4.3250346e-03  1.5538337e-02  1.1189972e-02
 3.2616558e-03  5.4309135e-03  4.6428950e-03 -4.0348615e-03]
```

Word 'doctor' not found in the Word2Vec model.
 Word 'patient' not found in the Word2Vec model.
 Vector embedding for 'guilty':

```
[-1.02168610e-02  1.05041070e-02 -8.49996414e-03 -1.94422179e-03
 -5.53654833e-03 -6.45571714e-03 -6.18769415e-03 -4.87722317e-03
 -6.03917427e-03 -3.62089323e-03 -3.43168178e-03  3.20895202e-03
 -9.64867231e-03 -3.50166997e-03  9.16745421e-03  5.36390767e-03
 1.19203934e-02  4.62551601e-03 -8.10001744e-04 -1.11547159e-02
 8.10768548e-03  2.89969891e-03  1.24709895e-02  4.68458841e-03
 8.06306209e-03  9.76778939e-03 -1.18508097e-02 -6.07924769e-03
 5.89404299e-05  7.53602665e-03  1.01619773e-02  3.62660550e-03
 1.07385321e-02 -1.03387106e-02  5.69567038e-03 -3.02940770e-03
 -8.49948078e-03  9.72383376e-03 -1.07314729e-03 -4.13596397e-03
 -3.01476498e-03 -4.96729976e-03  4.02944162e-03  4.84984741e-03
 -1.59257709e-03 -1.06038647e-02 -5.13104303e-03 -2.24396517e-03
 7.74389552e-03 -6.57061348e-04  4.25674880e-05  1.02711310e-04
 -7.99986534e-03  1.89318997e-03 -2.30359822e-03  2.23088288e-03
 6.83063269e-03 -2.04862049e-03 -4.20433003e-03  7.79549545e-03
 4.44924599e-03  4.12474712e-03  8.29528179e-03  1.06085688e-02
 4.05752333e-03  3.03117442e-03  7.49501260e-03  7.09697045e-03
 1.20756729e-03  4.30734188e-04 -5.46405441e-04 -2.02962221e-03
 -5.75965503e-03  4.27962560e-03  1.11803431e-02  8.60578753e-03
 -4.77114227e-04  8.38979147e-03  1.79873174e-03 -5.18136518e-03
 -2.62534944e-03 -6.45347033e-03  7.84642529e-03  8.15479085e-03
 2.72951415e-03  2.52984231e-03  1.21485880e-02  7.55486311e-03
 -3.63933941e-04 -6.03571860e-03  9.85072087e-03  7.77063146e-03
 2.89483299e-03 -2.88170367e-03 -2.65515177e-03 -2.94280332e-03
 9.14086122e-03 -4.40103095e-03 -7.24139577e-03  1.10059138e-02]
```

Word 'surgery' not found in the Word2Vec model.

```
In [6]: # Visualization using PCA
selected_words = ["court", "plaintiff", "defendant", "guilty", "jury",
                  "patient", "doctor", "hospital", "surgery", "emergency"]
word_vectors = [domain_word2vec.wv[word] for word in selected_words]

word_vectors
```

```
Out[6]: [array([-0.00520213,  0.05436571,  0.0196009 ,  0.00766893,  0.04851889,
 -0.22194375,  0.15068555,  0.2671535 , -0.16717364, -0.04062838,
 -0.054865 , -0.17729442, -0.06285486,  0.16066416,  0.00799252,
 0.00430546, -0.04130681, -0.11852198, -0.11586928, -0.32001996,
 0.07377547,  0.00634967,  0.01555517, -0.04018658, -0.05180506,
 -0.06574838,  0.01809591, -0.04998898, -0.05094941,  0.00987862,
 0.17092119, -0.03111312,  0.12419216, -0.07877786, -0.07952873,
 0.22328345,  0.12608306, -0.0951244 , -0.07667849, -0.1501351 ,
 0.04725789, -0.15457962, -0.06896634,  0.13114625,  0.11142956,
 0.03642106, -0.06946036, -0.02198208,  0.01422113,  0.05933676,
 0.09983439, -0.12603386,  0.07056595,  0.02597529, -0.02668819,
 0.0757888 , -0.00033602,  0.05289464, -0.16172495,  0.12800941,
 0.07429419,  0.10103885,  0.08504409, -0.01794797, -0.06241613,
 0.14987893,  0.15474467,  0.18398537, -0.17408288,  0.13962157,
 -0.11823418,  0.09919562,  0.07957372, -0.05181967,  0.15559544,
 0.0681076 , -0.0985308 ,  0.02557893, -0.11090399, -0.02128516,
 -0.01085772,  0.11211726, -0.14611867,  0.20995773, -0.10311343,
 0.06910679,  0.14604773,  0.10655196,  0.10023539, -0.02284993,
 0.14183174,  0.13799591,  0.00409749,  0.11127966,  0.21348046,
 0.03055387,  0.11364785, -0.1445034 ,  0.11242675, -0.04190433],
 dtype=float32),
 array([-0.03223411,  0.06478627,  0.00088969, -0.00806353,  0.05694845,
 -0.21240263,  0.13640128,  0.26523107, -0.13281158, -0.04770363,
 -0.02368818, -0.1402928 , -0.03685566,  0.12257947,  0.00039671,
 0.00741028, -0.01043882, -0.11464308, -0.09540985, -0.3000543 ,
 0.0647751 ,  0.00074026,  0.00411286, -0.05273201, -0.02684729,
 -0.04762366,  0.02497391, -0.04300669, -0.04396778, -0.00184753,
 0.14383827, -0.04924785,  0.08860843, -0.08550214, -0.06152922,
 0.24551614,  0.10724474, -0.13455397, -0.05984696, -0.15700217,
 0.02755019, -0.14089336, -0.07535081,  0.0659988 ,  0.11539416,
 0.020872 , -0.05348673, -0.02727061,  0.01346072,  0.03318129,
 0.09382757, -0.10529419,  0.0414049 ,  0.07656677, -0.01830849,
 0.07164428,  0.01196256,  0.05545417, -0.13542365,  0.1291954 ,
 0.08052401,  0.06550701,  0.09594982, -0.03788032, -0.07346537,
 0.16846505,  0.13681169,  0.14530386, -0.15170906,  0.14640196,
 -0.09068518,  0.0789521 ,  0.05557212, -0.02400086,  0.11684093,
 0.06631403, -0.11164055,  0.01440321, -0.10535935, -0.00458972,
 -0.02664629,  0.1090111 , -0.12968238,  0.18052402, -0.09392222,
 0.08443088,  0.12474449,  0.09482376,  0.11001488, -0.01367659,
 0.12273199,  0.1101999 ,  0.02236929,  0.09491293,  0.19617565,
 0.01282949,  0.11568122, -0.1593218 ,  0.10664962, -0.04113806],
 dtype=float32),
 array([ 0.00656709,  0.07256435, -0.0084228 , -0.02586134,  0.07641555,
 -0.2732658 ,  0.1540303 ,  0.32865882, -0.17496191, -0.06661771,
 -0.06085587, -0.22411431, -0.08474998,  0.18789086, -0.02127556,
```

```

0.03096173, -0.05577651, -0.12937057, -0.11135948, -0.36175218,
0.04432205, 0.00878906, 0.02296932, -0.05328603, -0.07712711,
-0.06075291, 0.04381331, -0.10575836, -0.06409874, 0.04152325,
0.21431115, -0.08531993, 0.14578514, -0.11424538, -0.11725931,
0.29418284, 0.10676998, -0.15401532, -0.09160217, -0.16645099,
0.06118093, -0.19756706, -0.08612581, 0.06556768, 0.1085471 ,
0.04415575, -0.06776308, -0.04802901, 0.04284215, 0.0638606 ,
0.13356939, -0.17036071, 0.08767819, 0.10464148, -0.03167466,
0.11624619, 0.03233933, 0.01407737, -0.15678538, 0.10963659,
0.11469187, 0.07420997, 0.09665452, -0.03110271, -0.07621247,
0.17188032, 0.18252161, 0.14339091, -0.22514871, 0.18041831,
-0.12061799, 0.07872933, 0.06301736, -0.04593184, 0.16801536,
0.08114434, -0.13756713, 0.06104114, -0.13863237, 0.01738747,
-0.01658883, 0.13528904, -0.1735411 , 0.24808215, -0.17541201,
0.04516907, 0.11772847, 0.14438275, 0.12152614, -0.00914207,
0.16980448, 0.15530077, -0.0296784 , 0.13635741, 0.24644977,
0.03516944, 0.11169897, -0.21215999, 0.10724142, -0.03329436],
dtype=float32),
array([-0.01413389, 0.06656995, -0.00734866, -0.03095385, 0.06509437,
-0.2517697 , 0.14954449, 0.29895368, -0.15728544, -0.07182206,
-0.06310162, -0.20050046, -0.08547995, 0.15693647, -0.0186175 ,
0.01778842, -0.05446635, -0.12549472, -0.11124176, -0.31952748,
0.03580405, 0.01365704, 0.03395955, -0.03605738, -0.06030127,
-0.04814158, 0.03859452, -0.09555041, -0.05513439, 0.0372526 ,
0.19865839, -0.07835107, 0.10888778, -0.11142128, -0.10577497,
0.29005775, 0.11180676, -0.13126965, -0.07538164, -0.1596524 ,
0.06402622, -0.17310387, -0.09087672, 0.04137763, 0.09426072,
0.02597058, -0.06627226, -0.02641308, 0.03379544, 0.0561525 ,
0.13159601, -0.16362782, 0.08867155, 0.10736878, -0.04391972,
0.10295371, 0.04891674, 0.00565069, -0.163432 , 0.08589575,
0.1108232 , 0.05997586, 0.11241774, -0.04420831, -0.06642649,
0.15975468, 0.1490166 , 0.12801382, -0.21193038, 0.1502985 ,
-0.10489336, 0.09517636, 0.0673286 , -0.03900745, 0.15302955,
0.0800889 , -0.13577344, 0.05731111, -0.12092727, 0.00424497,
-0.00455176, 0.11054221, -0.15298396, 0.20722686, -0.15278348,
0.03610937, 0.10936919, 0.14354476, 0.09363212, -0.00813364,
0.1714467 , 0.15730394, -0.02156785, 0.11239511, 0.24912179,
0.03659537, 0.0892475 , -0.202413 , 0.11249497, -0.05155509],
dtype=float32),
array([-0.00793692, 0.04061861, -0.01272589, -0.0216382 , 0.05832693,
-0.20959468, 0.1335544 , 0.23678838, -0.1236183 , -0.03556946,
-0.02290245, -0.1449683 , -0.06467045, 0.1215817 , -0.01873406,
0.01126286, -0.01548086, -0.10774158, -0.09506714, -0.26566857,
0.04896098, -0.00084279, 0.01825006, -0.05337542, -0.03144738,
-0.05701503, 0.03505556, -0.05904163, -0.04336127, -0.00061382,

```

Experiment 3

```

0.16488056, -0.05326047, 0.09773479, -0.07977082, -0.06476859,
0.22621374, 0.09030687, -0.11633091, -0.06397521, -0.13930038,
0.04200654, -0.13733749, -0.06120953, 0.06062739, 0.0891199 ,
0.02058195, -0.06635275, -0.00753717, 0.01779128, 0.05422449,
0.10981765, -0.1124575 , 0.06291748, 0.08702539, -0.03922568,
0.08803029, 0.03010272, 0.03673965, -0.13025954, 0.10378475,
0.07686505, 0.06131725, 0.09677017, -0.02306653, -0.05867948,
0.14983074, 0.11522046, 0.13194208, -0.16845842, 0.130707 ,
-0.09688476, 0.0888531 , 0.05799359, -0.03768088, 0.1355294 ,
0.04766061, -0.1006932 , 0.02873053, -0.10280664, -0.01355723,
-0.0322897 , 0.125441 , -0.13379173, 0.1888993 , -0.1090064 ,
0.05351871, 0.11118538, 0.09934786, 0.08427987, -0.01916844,
0.11553331, 0.12629525, -0.00194136, 0.09477089, 0.19319633,
0.01517526, 0.08618706, -0.15499583, 0.09854038, -0.04697568],
dtype=float32),
array([ 0.00135616, 0.06625096, 0.02714886, -0.03324671, 0.05406597,
-0.2076351 , 0.14450136, 0.27830392, -0.1474757 , -0.05214735,
-0.02860676, -0.218962 , -0.05803476, 0.11022121, -0.03196976,
0.0245685 , 0.0070367 , -0.12605277, -0.11396559, -0.3183468 ,
0.07659787, 0.01132763, 0.00593386, -0.04407553, -0.05708291,
-0.05022431, 0.03657781, -0.05108569, -0.0220301 , 0.00680075,
0.14817646, -0.03874053, 0.13069744, -0.11300313, -0.10196024,
0.2306353 , 0.13352849, -0.12474146, -0.07811124, -0.14196448,
0.03165774, -0.15317255, -0.04029788, 0.10843351, 0.11978162,
0.03644174, -0.07184896, -0.00125591, 0.01996329, 0.04686815,
0.12031849, -0.13361286, 0.07784432, 0.03898075, -0.05535794,
0.07788541, 0.02375661, 0.06319185, -0.13593689, 0.13807625,
0.04011758, 0.07736681, 0.10920981, -0.01097703, -0.08413535,
0.1694132 , 0.1142689 , 0.17812304, -0.16391632, 0.13841556,
-0.08013699, 0.09719803, 0.07872047, -0.04311903, 0.14359443,
0.06323478, -0.05998136, 0.03068179, -0.10644887, 0.00854869,
-0.04508544, 0.13762434, -0.12336963, 0.1855616 , -0.11391655,
0.09752344, 0.1405091 , 0.12214459, 0.11253129, -0.01929942,
0.13898279, 0.15566415, 0.01292162, 0.08838749, 0.19901091,
0.03416261, 0.12509196, -0.13636002, 0.11566975, -0.02010318],
dtype=float32),
array([-3.76006439e-02, 8.11468363e-02, -1.18198330e-02, 1.22082625e-02,
5.35595044e-03, -2.21441105e-01, 1.31108329e-01, 3.10447901e-01,
-2.11071640e-01, 7.52886664e-03, -6.67306557e-02, -1.76628768e-01,
-4.83631082e-02, 1.88437983e-01, -2.80619003e-02, 3.20329741e-02,
-2.19840016e-02, -1.36392176e-01, -1.02166705e-01, -3.58890593e-01,
4.39012572e-02, 4.81801666e-03, 1.11632412e-02, -6.98464885e-02,
-4.50425185e-02, -4.01994735e-02, -6.03534980e-04, -7.15099052e-02,
-7.36634061e-02, 2.14629583e-02, 2.10165456e-01, -6.25279024e-02,
1.19931854e-01, -1.26935437e-01, -8.21741298e-02, 2.74210095e-01,

```

Experiment 3

```

9.49538499e-02, -1.17289513e-01, -9.49264839e-02, -1.75545543e-01,
3.37264240e-02, -2.08480164e-01, -8.98559391e-02, 1.35834515e-01,
1.21459514e-01, 5.26671447e-02, -7.85357356e-02, -1.38883330e-02,
3.44770006e-03, 5.95685691e-02, 1.30519092e-01, -1.28386602e-01,
9.01534930e-02, 7.31256530e-02, -1.94634255e-02, 1.17376871e-01,
1.67697188e-04, 4.33479100e-02, -1.57258630e-01, 1.38467610e-01,
8.46170783e-02, 7.77027458e-02, 8.34437460e-02, -2.43678018e-02,
-8.29226896e-02, 1.89361051e-01, 1.67503580e-01, 2.07188442e-01,
-1.92358971e-01, 1.90954044e-01, -8.66395757e-02, 8.63512680e-02,
8.16990361e-02, -2.30716318e-02, 1.48350254e-01, 9.33871120e-02,
-1.03444301e-01, 3.32759172e-02, -1.03499167e-01, 2.95007881e-02,
-4.18480560e-02, 1.48850128e-01, -1.25358477e-01, 2.33333096e-01,
-1.20942295e-01, 1.06142171e-01, 1.28692985e-01, 1.23203449e-01,
1.00113675e-01, -1.41250789e-02, 1.63177848e-01, 1.50014937e-01,
-1.95683893e-02, 1.19940504e-01, 2.54336447e-01, 2.12510210e-02,
1.35626718e-01, -1.89367294e-01, 1.02768317e-01, -7.30541497e-02],
dtype=float32),
array([-0.02265889,  0.05778723, -0.01179005, -0.00284239,  0.04882376,
       -0.23377152,  0.13176689,  0.3085964 , -0.16563822, -0.00594464,
      -0.06069683, -0.16251391, -0.04316762,  0.19339406, -0.00984798,
       0.00349556, -0.03423214, -0.15440044, -0.15881209, -0.36713216,
       0.0536154 , -0.02835809, -0.01769544, -0.04901092, -0.05845137,
      -0.06207271,  0.01982044, -0.07527879, -0.0646024 ,  0.04246312,
       0.17291646, -0.03934861,  0.15174052, -0.11018316, -0.1073219 ,
       0.25728288,  0.10440008, -0.15727909, -0.0763182 , -0.1741864 ,
       0.02225032, -0.2151592 , -0.09898599,  0.08515406,  0.14156315,
       0.04146844, -0.080073 , -0.02897993,  0.03221606,  0.05149914,
       0.13052906, -0.13760065,  0.0776631 ,  0.07678478, -0.00785946,
       0.10275181, -0.01308092,  0.06987558, -0.16492371,  0.15031946,
       0.07514932,  0.06996216,  0.08479748,  0.00843247, -0.08604642,
       0.19633524,  0.18013164,  0.14563482, -0.19318359,  0.18252854,
      -0.10180707,  0.07443867,  0.04677813, -0.07124216,  0.16272344,
       0.05405647, -0.0953992 ,  0.04183496, -0.10839124, -0.01641163,
      -0.00933946,  0.11554954, -0.13658553,  0.22137882, -0.13358647,
       0.06070266,  0.13194123,  0.11046906,  0.12901476, -0.01263191,
       0.16108233,  0.17001428,  0.02732584,  0.10106397,  0.21696223,
       0.00993073,  0.13067529, -0.19392748,  0.11318995, -0.02350748],
dtype=float32),
array([-3.12990844e-02,  6.58327192e-02,  2.85430159e-03,  1.10345073e-02,
       -7.04743201e-03, -2.36223593e-01,  1.33402810e-01,  3.03116202e-01,
      -2.05681935e-01,  6.48758421e-03, -8.28733593e-02, -1.69779241e-01,
      -5.81854694e-02,  1.80510432e-01, -4.00698669e-02,  3.47116366e-02,
      -1.62971541e-02, -1.29537463e-01, -9.92213637e-02, -3.68670791e-01,
       4.55319285e-02,  8.06765445e-03, -1.78291200e-04, -6.00495152e-02,
      -5.73267005e-02, -4.28762138e-02, -3.84912407e-03, -6.40033185e-02,

```

```

-7.08072856e-02, 4.36537573e-03, 2.26468816e-01, -4.98397388e-02,
1.30335823e-01, -1.16139121e-01, -8.42535719e-02, 2.86336660e-01,
1.00505255e-01, -1.20256521e-01, -9.17292535e-02, -1.76113561e-01,
2.96843071e-02, -2.00398415e-01, -9.28441510e-02, 1.45912632e-01,
1.11865871e-01, 5.49624115e-02, -6.89490139e-02, -1.83873083e-02,
-1.00601949e-02, 6.59109801e-02, 1.25353217e-01, -1.26397550e-01,
9.62558836e-02, 5.71697466e-02, -2.06405111e-02, 1.16529934e-01,
-8.17977940e-04, 2.92389747e-02, -1.62125885e-01, 1.34710684e-01,
6.75722361e-02, 8.40188041e-02, 8.42126012e-02, -1.94504112e-02,
-1.00880139e-01, 1.89215228e-01, 1.60290688e-01, 2.12331533e-01,
-2.03707144e-01, 2.01542258e-01, -9.25249755e-02, 9.14819315e-02,
8.59961137e-02, -2.71495730e-02, 1.61703631e-01, 9.22792554e-02,
-1.11497119e-01, 5.09562343e-02, -1.00743666e-01, 3.40460427e-02,
-5.15895225e-02, 1.68939248e-01, -1.28210068e-01, 2.49226272e-01,
-1.33621320e-01, 1.16187118e-01, 1.42963469e-01, 1.47219375e-01,
1.09663606e-01, 5.80039807e-03, 1.60661057e-01, 1.45263568e-01,
-1.83158442e-02, 1.16535008e-01, 2.47885883e-01, 1.26237087e-02,
1.36337191e-01, -1.75651938e-01, 1.01963326e-01, -7.20273107e-02],
dtype=float32),
array([-0.01135001, 0.05962004, 0.03272124, -0.01768819, 0.04998965,
-0.20870571, 0.14685056, 0.2836007, -0.18323691, -0.03109219,
-0.03263121, -0.22973996, -0.04514541, 0.15115191, -0.02573079,
0.00774679, -0.00233633, -0.1304845, -0.14386515, -0.31636477,
0.06311441, 0.00088838, -0.02058102, -0.03525062, -0.05741948,
-0.08361816, 0.04948161, -0.06476792, -0.02724299, 0.00202177,
0.18204965, -0.03211843, 0.15414716, -0.11622933, -0.12767726,
0.25150353, 0.13327569, -0.16982682, -0.09414072, -0.16327456,
0.01518478, -0.16644533, -0.0390787, 0.10758833, 0.1320721,
0.01700985, -0.06482255, -0.01300713, 0.01951688, 0.04992113,
0.12908201, -0.15448081, 0.05776305, 0.0506245, -0.04097727,
0.08827188, 0.02562736, 0.04502805, -0.13268901, 0.15970598,
0.0415643, 0.10894849, 0.1138011, -0.03124123, -0.09126465,
0.17592564, 0.11290699, 0.18183088, -0.17974737, 0.15896654,
-0.07602367, 0.10534283, 0.06435065, -0.05782789, 0.17650633,
0.06925058, -0.07527371, 0.04818593, -0.11926682, -0.00753901,
-0.03426384, 0.13999003, -0.1504484, 0.19165526, -0.14357014,
0.09853069, 0.1328372, 0.13577645, 0.12897931, -0.0323601,
0.11762244, 0.17293827, -0.00854158, 0.07778574, 0.22550419,
0.03845395, 0.1585336, -0.14676552, 0.14424598, -0.03719835],  

dtype=float32)]
```

```
n [7]: pca = PCA(n_components=2)
reduced_vectors = pca.fit_transform(word_vectors)

reduced_vectors
```

```
Out[7]: array([[-0.06908131, -0.0287464 ],
 [-0.14953919,  0.01964696],
 [ 0.16693931,  0.13033459],
 [ 0.0674262 ,  0.16882876],
 [-0.1574409 ,  0.07644414],
 [-0.11961383, -0.04956438],
 [ 0.1142199 , -0.10011148],
 [ 0.06259862, -0.0240761 ],
 [ 0.12848931, -0.12535115],
 [-0.04399811, -0.06740492]])
```

```
In [8]: plt.figure(figsize=(12, 8))
for i, word in enumerate(selected_words):
    plt.scatter(reduced_vectors[i, 0], reduced_vectors[i, 1])
    plt.text(reduced_vectors[i, 0] + 0.002, reduced_vectors[i, 1], word, fontsize=12)
plt.title("PCA Visualization of Legal and Medical Word Embeddings")
plt.xlabel("PCA Dimension 1")
plt.ylabel("PCA Dimension 2")
plt.show()
```

Program 4

Use word embeddings to improve prompts for Generative AI model. Retrieve similar words using word embeddings. Use the similar words to enrich a GenAI prompt. Use the AI model to generate responses for the original and enriched prompts. Compare the outputs in terms of detail

In [2]: pip install transformers -U

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: transformers in c:\users\student\appdata\roaming\python\python311\site-packages (4.48.3)
Could not fetch URL https://pypi.org/simple/transformers/: There was a problem confirming the ssl certificate: HTTPSConnectionPool(host='pypi.org', port=443): Max retries exceeded with url: /simple/transformers/ (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self-signed certificate in certificate chain (_ssl.c:1006)'))) - skipping
Requirement already satisfied: filelock in c:\programdata\anaconda3\lib\site-packages (from transformers) (3.9.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.24.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from transformers) (0.28.1)
Requirement already satisfied: numpy>=1.17 in c:\users\student\appdata\roaming\python\python311\site-packages (from transformers) (1.24.4)
Requirement already satisfied: packaging>=20.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in c:\programdata\anaconda3\lib\site-packages (from transformers) (6.0)
Requirement already satisfied: regex!=2019.12.17 in c:\programdata\anaconda3\lib\site-packages (from transformers) (2022.7.9)
Requirement already satisfied: requests in c:\programdata\anaconda3\lib\site-packages (from transformers) (2.31.0)
Requirement already satisfied: tokenizers<0.22,>=0.21 in c:\users\student\appdata\roaming\python\python311\site-packages (from transformers) (0.21.0)
Requirement already satisfied: safetensors>=0.4.1 in c:\users\student\appdata\roaming\python\python311\site-packages (from transformers) (0.5.2)
Requirement already satisfied: tqdm>=4.27 in c:\programdata\anaconda3\lib\site-packages (from transformers) (4.65.0)
Requirement already satisfied: fsspec>=2023.5.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from huggingface-hub<1.0,>=0.24.0->transformers) (2025.2.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\student\appdata\roaming\python\python311\site-packages (from huggingface-hub<1.0,>=0.24.0->transformers) (4.12.2)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-packages (from tqdm>=4.27->transformers) (0.4.6)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests->transformers) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests->transformers) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\student\appdata\roaming\python\python311\site-packages (from requests->transformers) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests->transformers) (2023.7.22)
Could not fetch URL https://pypi.org/simple/pip/: There was a problem confirming the ssl certificate: HTTPSConnectionPool(host='pypi.org', port=443): Max retries exceeded with url: /simple/pip/ (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self-signed certificate in certificate chain (_ssl.c:1006)'))) - skipping
Note: you may need to restart the kernel to use updated packages.
```

Activate |
Go to Settings

```
WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSL Error(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self-signed certificate in certificate chain (_ssl.c:1006)'))': /simple/transfomers/
WARNING: Retrying (Retry(total=3, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSL Error(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self-signed certificate in certificate chain (_ssl.c:1006)'))': /simple/transfomers/
WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSL Error(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self-signed certificate in certificate chain (_ssl.c:1006)'))': /simple/transfomers/
WARNING: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSL Error(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self-signed certificate in certificate chain (_ssl.c:1006)'))': /simple/transfomers/
WARNING: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSL Error(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self-signed certificate in certificate chain (_ssl.c:1006)'))': /simple/transfomers/
```

```
In [1]: from gensim.scripts.glove2word2vec import glove2word2vec
from gensim.models import KeyedVectors

# Paths to the GloVe file and output Word2Vec file
glove_input_file = "glove.6B/glove.6B.100d.txt" # Path to GloVe file
word2vec_output_file = "glove.6B/glove.6B.100d.word2vec.txt" # Output file in Word2Vec format

# Convert GloVe format to Word2Vec format
glove2word2vec(glove_input_file, word2vec_output_file)

# Load the converted Word2Vec model
model = KeyedVectors.load_word2vec_format(word2vec_output_file, binary=False)

# Test the loaded model
print(model.most_similar("king"))

C:\Users\student\AppData\Local\Temp\ipykernel_13328\2083156905.py:9: DeprecationWarning: Call to deprecated `glove2word2vec` (KeyedVectors.load_word2vec_format(..., binary=False, no_header=True) loads GLoVE text vectors.).
  glove2word2vec(glove_input_file, word2vec_output_file)
[('prince', 0.7682328820228577), ('queen', 0.7507690787315369), ('son', 0.7020888328552246), ('brother', 0.6985775232315063), ('monarch', 0.6977890729904175), ('throne', 0.6919989585876465), ('kingdom', 0.6811409592628479), ('father', 0.6802029013633728), ('emperor', 0.6712858080863953), ('ii', 0.6676074266433716)]
```

```
In [2]: # Define the original medical prompt
original_prompt = "Explain the importance of vaccinations in healthcare."

# Define key terms extracted from the original prompt
key_terms = ["vaccinations", "healthcare"]
```

Activat
Go to Set

```

# Initialize an empty list to store similar terms
similar_terms = []

# Loop through each key term to find similar words
for term in key_terms:
    # Check if the key term exists in the vocabulary of the 'model' (word embedding model)
    # Assuming 'model.key_to_index' is a way to check for term existence in the model's vocabulary
    if term in model.key_to_index:
        # If the term exists, find the top 3 most similar words using 'model.most_similar(term, topn=3)'
        # and extend the 'similar_terms' list with these words.
        # Assuming 'model.most_similar' returns a list of tuples, where each tuple is (word, similarity_score)
        # We are extracting only the 'word' part using a set comprehension for potential deduplication.
        similar_terms.extend({word for word, _ in model.most_similar(term, topn=3)})

# Enrich the original prompt with the retrieved similar terms
if similar_terms:
    # If similar terms were found, create an enriched prompt by appending
    # "Consider aspects like: " followed by a comma-separated string of similar terms.
    enriched_prompt = f"{original_prompt} Consider aspects like: {', '.join(similar_terms)}."
else:
    # If no similar terms were found, the enriched prompt is the same as the original prompt.
    enriched_prompt = original_prompt

# Output the original and enriched prompts
print("Original Prompt:", original_prompt)
print("Enriched Prompt:", enriched_prompt)

```

Original Prompt: Explain the importance of vaccinations in healthcare.

Enriched Prompt: Explain the importance of vaccinations in healthcare. Consider aspects like: inoculations, vaccinations, immunizations, health, care, services.

```
In [3]: import getpass
import os
GOOGLE_API_KEY= os.environ["GOOGLE_API_KEY"] = getpass.getpass("Enter your Google AI API key: ")
```

Enter your Google AI API key:

```
In [4]: from langchain_google_genai import ChatGoogleGenerativeAI

llm = ChatGoogleGenerativeAI(
    model="gemini-2.0-flash-exp",
    temperature=0,
    api_key=GOOGLE_API_KEY,
    max_tokens=256,
    timeout=None,
    max_retries=2,
```

Activate
Go to Seti

```

        # other params...
    )

In [5]: llm.invoke("Hi")

Out[5]: AIMessage(content='Hi there! How can I help you today?', additional_kwargs={}, response_metadata={'prompt_feedback':
{'block_reason': 0, 'safety_ratings': []}, 'finish_reason': 'STOP', 'safety_ratings': []}, id='run-a92f789f-c695-4dc5-9
f90-ee8c2f9bb121-0', usage_metadata={'input_tokens': 1, 'output_tokens': 11, 'total_tokens': 12, 'input_token_details':
{'cache_read': 0}})

In [6]: print(llm.invoke(original_prompt).content)

Vaccinations are a cornerstone of modern healthcare and play a vital role in protecting individuals and communities from infectious diseases. Their importance can be summarized in several key areas:

**1. Disease Prevention and Eradication:**

* **Individual Protection:** Vaccines work by exposing the body to a weakened or inactive form of a disease-causing agent (virus or bacteria). This triggers the immune system to produce antibodies, which provide protection against future infections. If the individual is later exposed to the real disease, their immune system is primed to fight it off quickly and effectively, often preventing illness or reducing its severity.
* **Herd Immunity:** When a large percentage of a population is vaccinated, it creates "herd immunity." This means that even those who cannot be vaccinated (e.g., infants too young, individuals with certain medical conditions) are protected because the disease has difficulty spreading. Herd immunity is crucial for protecting vulnerable populations.
* **Disease Eradication/Elimination:** Vaccination campaigns have successfully eradicated diseases like smallpox and have significantly reduced the incidence of others, such as polio and measles. Continued vaccination efforts are essential to maintain these achievements and prevent resurgence.

**2. Reduced Morbidity and Mortality:**

* **Preventing**

```

In [9]: print(llm.invoke(enriched_prompt).content)

The Vital Role of Vaccinations in Healthcare

Vaccinations are a cornerstone of modern healthcare, playing a crucial role in preventing infectious diseases and promoting overall health and well-being. They are a prime example of preventative medicine, offering significant benefits at both the individual and population levels. Let's break down their importance considering the aspects you mentioned:

1. Inoculations, Vaccinations, and Immunizations: Understanding the Terms*

- * **Inoculation:** Historically, this term referred to introducing a small amount of infectious material (like pus from a smallpox sore) into a healthy person to induce a mild infection and subsequent immunity. It was a precursor to modern vaccination.
- * **Vaccination:** This is the process of administering a vaccine, a biological preparation that provides active acquired immunity to a particular infectious disease. Vaccines typically contain weakened or inactive forms of the disease-causing agent, or parts of it.
- * **Immunization:** This refers to the process by which a person becomes protected against a disease, either through vaccination or natural infection. Vaccination is a key method of achieving immunization.

2. Health and Care: The Direct Impact of Vaccinations*

- * **Individual Health:** Vaccinations protect individuals from contracting serious and

Program 5

Use word embeddings to create meaningful sentences for creative tasks. Retrieve similar words for a seed word.
 Create a sentence or story using these words as a starting point. Write a program that: Takes a seed word. Generates similar words. Constructs a short paragraph using these words.

```
In [1]: !pip install sentence_transformers
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: sentence_transformers in c:\users\student\appdata\roaming\python\python311\site-packages
(3.4.1)
Requirement already satisfied: transformers<5.0.0,>=4.41.0 in c:\users\student\appdata\roaming\python\python311\site-packages
(from sentence_transformers) (4.48.3)
Requirement already satisfied: tqdm in c:\programdata\anaconda3\lib\site-packages (from sentence_transformers) (4.65.0)
Requirement already satisfied: torch>=1.11.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from s
entence_transformers) (2.6.0)
Requirement already satisfied: scikit-learn in c:\programdata\anaconda3\lib\site-packages (from sentence_transformers)
(1.3.0)
Requirement already satisfied: scipy in c:\users\student\appdata\roaming\python\python311\site-packages (from sentence_
transformers) (1.10.1)
Requirement already satisfied: huggingface-hub>=0.20.0 in c:\users\student\appdata\roaming\python\python311\site-packag
es (from sentence_transformers) (0.28.1)
Requirement already satisfied: Pillow in c:\programdata\anaconda3\lib\site-packages (from sentence_transformers) (9.4.
0)
Requirement already satisfied: filelock in c:\programdata\anaconda3\lib\site-packages (from huggingface-hub>=0.20.0->se
ntence_transformers) (3.9.0)
Requirement already satisfied: fsspec>=2023.5.0 in c:\users\student\appdata\roaming\python\python311\site-packages (fro
m huggingface-hub>=0.20.0->sentence_transformers) (2025.2.0)
Requirement already satisfied: packaging>=20.9 in c:\users\student\appdata\roaming\python\python311\site-packages (from
huggingface-hub>=0.20.0->sentence_transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in c:\programdata\anaconda3\lib\site-packages (from huggingface-hub>=0.20.0-
>sentence_transformers) (6.0)
Requirement already satisfied: requests in c:\programdata\anaconda3\lib\site-packages (from huggingface-hub>=0.20.0->se
ntence_transformers) (2.31.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\student\appdata\roaming\python\python311\site-pac
kages (from huggingface-hub>=0.20.0->sentence_transformers) (4.12.2)
Requirement already satisfied: networkx in c:\programdata\anaconda3\lib\site-packages (from torch>=1.11.0->sentence_tr
ansformers) (3.1)
Requirement already satisfied: jinja2 in c:\programdata\anaconda3\lib\site-packages (from torch>=1.11.0->sentence_trans
formers) (3.1.2)
Requirement already satisfied: sympy==1.13.1 in c:\users\student\appdata\roaming\python\python311\site-packages (from t
orch>=1.11.0->sentence_transformers) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\programdata\anaconda3\lib\site-packages (from sympy==1.13.1->t
orch>=1.11.0->sentence_transformers) (1.3.0)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-packages (from tqdm->sentence_transformer
s) (0.4.6)
Requirement already satisfied: numpy>=1.17 in c:\users\student\appdata\roaming\python\python311\site-packages (from tra
nsformers<5.0.0,>=4.41.0->sentence_transformers) (1.24.4)
Requirement already satisfied: regex!=2019.12.17 in c:\programdata\anaconda3\lib\site-packages (from transformers<5.0.
0,>=4.41.0->sentence_transformers) (2022.7.9)
Requirement already satisfied: tokenizers<0.22,>=0.21 in c:\users\student\appdata\roaming\python\python311\site-package
s (from transformers<5.0.0,>=4.41.0->sentence_transformers) (0.21.0)
Requirement already satisfied: safetensors>=0.4.1 in c:\users\student\appdata\roaming\python\python311\site-packages (f

```

```
rom transformers<5.0.0,>=4.41.0->sentence_transformers) (0.5.2)
Requirement already satisfied: joblib>=1.1.1 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn->sentence_transformers) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn->sentence_transformers) (2.2.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\programdata\anaconda3\lib\site-packages (from jinja2->torch>=1.11.0->sentence_transformers) (2.1.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests->huggingface-hub>=0.20.0->sentence_transformers) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests->huggingface-hub>=0.20.0->sentence_transformers) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\student\appdata\roaming\python\python311\site-packages (from requests->huggingface-hub>=0.20.0->sentence_transformers) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests->huggingface-hub>=0.20.0->sentence_transformers) (2023.7.22)
```

In [1]: `!pip install langchain-huggingface`

```

Requirement already satisfied: regex!=2019.12.17 in c:\programdata\anaconda3\lib\site-packages (from transformers>=4.3
9.0->langchain-huggingface) (2022.7.9)
Requirement already satisfied: safetensors>=0.4.1 in c:\users\student\appdata\roaming\python\python311\site-packages (f
rom transformers>=4.39.0->langchain-huggingface) (0.5.2)
Requirement already satisfied: jsonpointer>=1.9 in c:\programdata\anaconda3\lib\site-packages (from jsonpatch<2.0,>=1.3
3->langchain-core<0.4.0,>=0.3.15->langchain-huggingface) (2.1)
Requirement already satisfied: httpx<1,>=0.23.0 in c:\users\student\appdata\roaming\python\python311\site-packages (fro
m langsmith<0.4,>=0.1.125->langchain-core<0.4.0,>=0.3.15->langchain-huggingface) (0.28.1)
Requirement already satisfied: orjson<4.0.0,>=3.9.14 in c:\users\student\appdata\roaming\python\python311\site-packages
(from langsmith<0.4,>=0.1.125->langchain-core<0.4.0,>=0.3.15->langchain-huggingface) (3.10.15)
Requirement already satisfied: requests_toolbelt<2.0.0,>=1.0.0 in c:\programdata\anaconda3\lib\site-packages (from lang
smith<0.4,>=0.1.125->langchain-core<0.4.0,>=0.3.15->langchain-huggingface) (1.0.0)
Requirement already satisfied: zstandard<0.24.0,>=0.23.0 in c:\users\student\appdata\roaming\python\python311\site-pac
kages (from langsmith<0.4,>=0.1.125->langchain-core<0.4.0,>=0.3.15->langchain-huggingface) (0.23.0)
Requirement already satisfied: annotated-types>=0.6.0 in c:\users\student\appdata\roaming\python\python311\site-package
s (from pydantic<3.0.0,>=2.5.2->langchain-core<0.4.0,>=0.3.15->langchain-huggingface) (0.7.0)
Requirement already satisfied: pydantic-core==2.27.2 in c:\users\student\appdata\roaming\python\python311\site-packages
(from pydantic<3.0.0,>=2.5.2->langchain-core<0.4.0,>=0.3.15->langchain-huggingface) (2.27.2)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests->h
uggingface-hub>=0.23.0->langchain-huggingface) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests->huggingface-h
ub>=0.23.0->langchain-huggingface) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\student\appdata\roaming\python\python311\site-packages (f
rom requests->huggingface-hub>=0.23.0->langchain-huggingface) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests->huggin
gface-hub>=0.23.0->langchain-huggingface) (2023.7.22)
Requirement already satisfied: networkx in c:\programdata\anaconda3\lib\site-packages (from torch>=1.11.0->sentence-tr
ansformers>=2.6.0->langchain-huggingface) (3.1)
Requirement already satisfied: jinja2 in c:\programdata\anaconda3\lib\site-packages (from torch>=1.11.0->sentence-trans
formers>=2.6.0->langchain-huggingface) (3.1.2)
Requirement already satisfied: sympy==1.13.1 in c:\users\student\appdata\roaming\python\python311\site-packages (from t
orch>=1.11.0->sentence-transformers>=2.6.0->langchain-huggingface) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\programdata\anaconda3\lib\site-packages (from sympy==1.13.1->t
orch>=1.11.0->sentence-transformers>=2.6.0->langchain-huggingface) (1.3.0)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-packages (from tqdm>=4.42.1->huggingface-h
ub>=0.23.0->langchain-huggingface) (0.4.6)
Requirement already satisfied: joblib>=1.1.1 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn->sentence
-transformers>=2.6.0->langchain-huggingface) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn->s
entence-transformers>=2.6.0->langchain-huggingface) (2.2.0)
Requirement already satisfied: anyio in c:\programdata\anaconda3\lib\site-packages (from httpx<1,>=0.23.0->langsmith<0
.4,>=0.1.125->langchain-core<0.4.0,>=0.3.15->langchain-huggingface) (3.5.0)
Requirement already satisfied: httpcore==1.* in c:\users\student\appdata\roaming\python\python311\site-packages (from h
ttpx<1,>=0.23.0->langsmith<0.4,>=0.1.125->langchain-core<0.4.0,>=0.3.15->langchain-huggingface) (1.0.7)          Activate Win
httpcore==1.*->httpx<1,>=0.23.0->langsmith<0.4,>=0.1.125->langchain-core<0.4.0,>=0.3.15->langchain-huggingface) (0.14.
0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\programdata\anaconda3\lib\site-packages (from jinja2->torch>=1.11.
0->sentence-transformers>=2.6.0->langchain-huggingface) (2.1.1)
Requirement already satisfied: sniffio>=1.1 in c:\programdata\anaconda3\lib\site-packages (from anyio->httpx<1,>=0.23.0
->langsmith<0.4,>=0.1.125->langchain-core<0.4.0,>=0.3.15->langchain-huggingface) (1.2.0)

```

In [1]: !pip install tf-keras --user

```
Requirement already satisfied: tf-keras in c:\users\student\appdata\roaming\python\python311\site-packages (2.18.0)
Requirement already satisfied: tensorflow<2.19,>=2.18 in c:\users\student\appdata\roaming\python\python311\site-packages (from tf-keras) (2.18.0)
Requirement already satisfied: tensorflow-intel==2.18.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow<2.19,>=2.18->tf-keras) (2.18.0)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (25.1.24)
Requirement already satisfied: gast!=0.5.0,!>0.5.1,!>0.5.2,>=0.2.1 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (3.4.0)
Requirement already satisfied: packaging in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (24.2)
Requirement already satisfied: protobuf!=4.21.0,!>4.21.1,!>4.21.2,!>4.21.3,!>4.21.4,!>4.21.5,<6.0.0dev,>=3.20.3 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (5.29.3)
Requirement already satisfied: requests<3,>=2.21.0 in c:\programdata\anaconda3\lib\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (2.31.0)
Requirement already satisfied: setuptools in c:\programdata\anaconda3\lib\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (68.0.0)
Requirement already satisfied: six>=1.12.0 in c:\programdata\anaconda3\lib\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in c:\programdata\anaconda3\lib\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (1.14.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (1.70.0)
Requirement already satisfied: tensorboard<2.19,>=2.18 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (2.18.0)
Requirement already satisfied: keras>=3.5.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (3.8.0)
Collecting numpy<2.1.0,>=1.26.0 (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras)
  Obtaining dependency information for numpy<2.1.0,>=1.26.0 from https://files.pythonhosted.org/packages/eb/57/3a3f14d3
```

```

Using cached numpy-2.0.2-cp311-cp311-win_amd64.whl.metadata (59 kB)
Requirement already satisfied: h5py>=3.11.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (3.12.1)
Requirement already satisfied: m1-dtypes<0.5.0,>=0.4.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (0.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (0.31.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\programdata\anaconda3\lib\site-packages (from astunparse>=1.6.0->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (0.38.4)
Requirement already satisfied: rich in c:\users\student\appdata\roaming\python\python311\site-packages (from keras>=3.5.0->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (13.9.4)
Requirement already satisfied: namex in c:\users\student\appdata\roaming\python\python311\site-packages (from keras>=3.5.0->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (0.0.8)
Requirement already satisfied: optree in c:\users\student\appdata\roaming\python\python311\site-packages (from keras>=3.5.0->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (0.14.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\student\appdata\roaming\python\python311\site-packages (from requests<3,>=2.21.0->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (2023.7.22)
Requirement already satisfied: markdown>=2.6.8 in c:\programdata\anaconda3\lib\site-packages (from tensorboard<2.19,>=2.18->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (3.4.1)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from tensorboard<2.19,>=2.18->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in c:\programdata\anaconda3\lib\site-packages (from tensorboard<2.19,>=2.18->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (2.2.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\programdata\anaconda3\lib\site-packages (from werkzeug>=1.0.1->tensorboard<2.19,>=2.18->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (2.1.1)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\programdata\anaconda3\lib\site-packages (from rich->keras>=3.5.0->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\programdata\anaconda3\lib\site-packages (from rich->keras>=3.5.0->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (2.15.1)
Requirement already satisfied: mdurl~0.1 in c:\programdata\anaconda3\lib\site-packages (from markdown-it-py>=2.2.0->rich->keras>=3.5.0->tensorflow-intel==2.18.0->tensorflow<2.19,>=2.18->tf-keras) (0.1.0)
Using cached numpy-2.0.2-cp311-cp311-win_amd64.whl (15.9 MB)

Installing collected packages: numpy
  Attempting uninstall: numpy
    Found existing installation: numpy 1.24.4
    Uninstalling numpy-1.24.4:
      Successfully uninstalled numpy-1.24.4

```

```
WARNING: The scripts f2py.exe and numpy-config.exe are installed in 'C:\Users\student\AppData\Roaming\Python\Python311\Scripts' which is not on PATH.
```

```
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
```

```
ERROR: Could not install packages due to an OSError: [WinError 5] Access is denied: 'C:\Users\student\AppData\Roaming\Python\Python311\site-packages\~mpy\.libs\libopenblas64_v0.3.21-gcc_10_3_0.dll'
```

```
Check the permissions.
```

```
In [1]: !pip install numpy==1.24.4 --user
```

```
Collecting numpy==1.24.4
```

```
  Obtaining dependency information for numpy==1.24.4 from https://files.pythonhosted.org/packages/d8/ec/ebef2f7d7c28503f958f0f8b992e7ce606fb74f9e891199329d5f5f87404/numpy-1.24.4-cp311-cp311-win_amd64.whl.metadata
```

```
    Using cached numpy-1.24.4-cp311-cp311-win_amd64.whl.metadata (5.6 kB)
```

```
Using cached numpy-1.24.4-cp311-cp311-win_amd64.whl (14.8 MB)
```

```
Installing collected packages: numpy
```

```
  Attempting uninstall: numpy
```

```
    Found existing installation: numpy 2.0.2
```

```
    Uninstalling numpy-2.0.2:
```

```
      Successfully uninstalled numpy-2.0.2
```

```
Successfully installed numpy-1.24.4
```

```
WARNING: The script f2py.exe is installed in 'C:\Users\student\AppData\Roaming\Python\Python311\Scripts' which is not on PATH.
```

```
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
```

```
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behavior is the source of the following dependency conflicts.
```

```
tables 3.8.0 requires blosc2~=2.0.0, which is not installed.
```

```
tables 3.8.0 requires cython>=0.29.21, which is not installed.
```

```
tensorflow-intel 2.18.0 requires numpy<2.1.0,>=1.26.0, but you have numpy 1.24.4 which is incompatible.
```

```
In [3]: !pip install --upgrade --quiet huggingface_hub
```

```
In [5]: pip install --upgrade urllib3
```

```
Defaulting to user installation because normal site-packages is not writeable
```

```
Requirement already satisfied: urllib3 in c:\users\student\appdata\roaming\python\python311\site-packages (2.3.0)
```

```
Note: you may need to restart the kernel to use updated packages.
```

```
In [6]: pip freeze
```

```
absl-py==2.1.0
aiobotocore @ file:///C:/b/abs_1c1a_vjay2/croot/aiobotocore_1682537737724/work
aiofiles @ file:///C:/b/abs_9ex6mi6b56/croot/aiofiles_1683773603390/work
aiohttp @ file:///C:/b/abs_b78zt6vo64/croot/aiohttp_1694181126607/work
aioitertools @ file:///tmp/build/80754af9/aioitertools_1607109665762/work
aiosignal @ file:///tmp/build/80754af9/aiosignal_1637843061372/work
aiosqlite @ file:///C:/b/abs_9djc_0pyi3/croot/aiosqlite_1683773915844/work
alabaster @ file:///home/ktietz/src/ci/alabaster_1611921544520/work
anaconda-anon-usage @ file:///C:/b/abs_f4tsjy19va/croot/anaconda-anon-usage_1695310457827/work
anaconda-catalogs @ file:///C:/b/abs_8btty0o8s8/croot/anaconda-catalogs_1685727315626/work
anaconda-client @ file:///C:/b/abs_80wttmgui4/croot/anaconda-client_1694625288614/work
anaconda-cloud-auth @ file:///C:/b/abs_5cjpn6wj6/croot/anaconda-cloud-auth_1694462130037/work
anaconda-navigator @ file:///C:/b/abs_ab00e0_u7e/croot/anaconda-navigator_1695238210954/work
anaconda-project @ file:///C:/ci_311/anaconda-project_1676458365912/work
annotated-types==0.7.0
anyio @ file:///C:/ci_311/anyio_1676425491996/work/dist
appdirs==1.4.4
argon2-cffi @ file:///opt/conda/conda-bld/argon2-cffi_1645000214183/work
argon2-cffi-bindings @ file:///C:/ci_311/argon2-cffi-bindings_1676424443321/work
arrow @ file:///C:/ci_311/arrow_1678249767083/work
astroid @ file:///C:/ci_311/astroid_1678740610167/work
astropy @ file:///C:/ci_311_rebuilds/astropy_1678996071858/work
asttokens @ file:///opt/conda/conda-bld/asttokens_1646925590279/work
astunparse==1.6.3
async-timeout @ file:///C:/ci_311/async-timeout_1676431518331/work
atomicwrites==1.4.0
attrs @ file:///C:/ci_311/attrs_1676422272484/work
Automat @ file:///tmp/build/80754af9/automat_1600298431173/work
autopep8 @ file:///opt/conda/conda-bld/autopep8_1650463822033/work
Babel @ file:///C:/ci_311/babel_1676427169844/work
backcall @ file:///home/ktietz/src/ci/backcall_1611930011877/work
backports.functools-lru-cache @ file:///tmp/build/80754af9/backports.functools_lru_cache_1618170165463/work
backports tempfile @ file:///home/linux1/recipes/ci/backports.tempfile_1610991236607/work
backports.weakref==1.0.post1
bcrypt @ file:///C:/ci_311/bcrypt_1676435170049/work
beautifulsoup4 @ file:///C:/b/abs_0agyz1wsr4/croot/beautifulsoup4-split_1681493048687/work
binaryornot @ file:///tmp/build/80754af9/binaryornot_1617751525010/work
black @ file:///C:/b/abs_620t6ndje8/croot/black_1680737261963/work
bleach @ file:///opt/conda/conda-bld/bleach_1641577558959/work
bokeh @ file:///C:/b/abs_e5qs_0dl2w/croot/bokeh_1690546119144/work
boltons @ file:///C:/ci_311/boltons_1677729932371/work
botocore @ file:///C:/b/abs_01gwdn34ju/croot/botocore_1682528022942/work
Bottleneck @ file:///C:/ci_311/bottleneck_1676500016583/work
brotliipy==0.7.0
cachetools==5.5.1
```



```
incremental @ file:///tmp/build/80754af9/incremental_1636629750599/work
inflection==0.5.1
iniconfig @ file:///home/linux1/recipes/ci/iniconfig_1610983019677/work
intake @ file:///C:/ci_311_rebuilds/intake_1678999914269/work
intervaltree @ file:///Users/ktietz/demo/mc3/conda-bld/intervaltree_1630511889664/work
ipykernel @ file:///C:/b/abs_07rkft_vaz/croot/ipykernel_1691121700587/work
ipython @ file:///C:/b/abs_e5729i179y/croot/ipython_1694181400005/work
ipython-genutils @ file:///tmp/build/80754af9/ipython_genutils_1606773439826/work
ipywidgets @ file:///C:/b/abs_5awapknmz_/croot/ipywidgets_1679394824767/work
isort @ file:///tmp/build/80754af9/isort_1628603791788/work
itemadapter @ file:///tmp/build/80754af9/itemadapter_1626442940632/work
itemloaders @ file:///opt/conda/conda-bld/itemloaders_1646805235997/work
itsdangerous @ file:///tmp/build/80754af9/itsdangerous_1621432558163/work
jaraco.classes @ file:///tmp/build/80754af9/jaraco.classes_1620983179379/work
jedi @ file:///C:/ci_311/jedi_1679427407646/work
jellyfish @ file:///C:/b/abs_50kgvtvrbj/croot/jellyfish_1695193564091/work
Jinja2 @ file:///C:/ci_311/jinja2_1676424968965/work
jinja2-time @ file:///opt/conda/conda-bld/jinja2-time_1649251842261/work
jmespath @ file:///Users/ktietz/demo/mc3/conda-bld/jmespath_1630583964805/work
joblib @ file:///C:/b/abs_1anqjntpan/croot/joblib_1685113317150/work
json5 @ file:///tmp/build/80754af9/json5_1624432770122/work
jsonpatch==1.33
jsonpointer==2.1
jsonschema @ file:///C:/b/abs_d40z05b6r1/croot/jsonschema_1678983446576/work
jupyter @ file:///C:/ci_311/jupyter_1678249952587/work
jupyter-console @ file:///C:/b/abs_82xaaa612y4/croot/jupyter_console_1680000189372/work
jupyter-events @ file:///C:/b/abs_4cak_28ewz/croot/jupyter_events_1684268050893/work
jupyter-server @ file:///C:/ci_311/jupyter_server_1678228762759/work
jupyter-ydoc @ file:///C:/b/abs_e7m6nh5lao/croot/jupyter_ydoc_1683747253535/work
jupyter_client @ file:///C:/ci_311/jupyter_client_1676424009414/work
jupyter_core @ file:///C:/b/abs_9d0ttho3bs/croot/jupyter_core_1679906581955/work
jupyter_server_fileid @ file:///C:/b/abs_f1yjnmiq_6/croot/jupyter_server_fileid_1684273602142/work
jupyter_server_ydoc @ file:///C:/b/abs_8ai39bligw/croot/jupyter_server_ydoc_1686767445888/work
jupyterlab @ file:///C:/b/abs_c1msr8zz3y/croot/jupyterlab_1686179674844/work
jupyterlab-pygments @ file:///tmp/build/80754af9/jupyterlab_pygments_1601490720602/work
jupyterlab-widgets @ file:///C:/b/abs_38ad427jkz/croot/jupyterlab_widgets_1679055289211/work
jupyterlab_server @ file:///C:/b/abs_e0qqsihjvl/croot/jupyterlab_server_1680792526136/work
kaleido @ file:///C:/b/abs_60smvjz1os/croot/python-kaleido_1689927138239/work
keras==3.8.0
keyring @ file:///C:/b/abs_dbjc7g0dh2/croot/keyring_1678999228878/work
kiwisolver @ file:///C:/ci_311/kiwisolver_1676431979301/work
langchain==0.3.17
langchain-cohere==0.4.2
langchain-community==0.3.16
langchain-core==0.3.33
```

```
langchain-google-genai==2.0.9
langchain-huggingface==0.1.2
langchain-text-splitters==0.3.5
langsmith==0.3.4
lazy-object-proxy @ file:///C:/ci_311/lazy-object-proxy_1676432050939/work
lazy_loader @ file:///C:/b/abs_c9j1w06oq1/croot/lazy_loader_1687266162676/work
libarchive-c @ file:///tmp/build/80754af9/python-libarchive-c_1617780486945/work
libclang==18.1.1
libmambapy @ file:///C:/b/abs_71g8gec0dd/croot/mamba-split_1694187821755/work/libmambapy
linkify-it-py @ file:///C:/ci_311/linkify-it-py_1676474436187/work
llvmlite @ file:///C:/b/abs_a8i9keuf6p/croot/llvmlite_1683555140340/work
lmdb @ file:///C:/b/abs_556ronuvb2/croot/python-lmdb_1682522366268/work
locket @ file:///C:/ci_311/loket_1676428325082/work
lxml @ file:///C:/b/abs_9e7tpg2vv9/croot/lxml_1695058219431/work
lz4 @ file:///C:/b/abs_064u6aszy3/croot/lz4_1686057967376/work
Markdown @ file:///C:/ci_311/markdown_1676437912393/work
markdown-it-py @ file:///C:/b/abs_a5bfngz6fu/croot/markdown-it-py_1684279915556/work
MarkupSafe @ file:///C:/ci_311/markupsafe_1676424152318/work
marshmallow==3.26.1
matplotlib @ file:///C:/b/abs_085jhivdha/croot/matplotlib-suite_1693812524572/work
matplotlib-inline @ file:///C:/ci_311/matplotlib-inline_1676425798036/work
mccabe @ file:///opt/conda/conda-bld/mccabe_1644221741721/work
mdit-py-plugins @ file:///C:/ci_311/mdit-py-plugins_1676481827414/work
mdurl @ file:///C:/ci_311/mdurl_1676442676678/work
menuinst @ file:///C:/ci_311/menuinst_1678730372782/work
miniflue==0.0.6
mistune @ file:///C:/ci_311/mistune_1676425149302/work
mkl-fft @ file:///C:/b/abs_19i1y8ykas/croot/mkl_fft_1695058226480/work
mkl-random @ file:///C:/b/abs_edwkj1_o69/croot/mkl_random_1695059866750/work
mkl-service==2.4.0
ml-dtypes==0.4.1
more-itertools @ file:///tmp/build/80754af9/more-itertools_1637733554872/work
mpmath @ file:///C:/b/abs_7833jrbiox/croot/mpmath_1690848321154/work
msgpack @ file:///C:/ci_311/msgpack-python_1676427482892/work
multidict @ file:///C:/ci_311/multidict_1676428396308/work
multipledispatch @ file:///C:/ci_311/multipledispatch_1676442767760/work
multiprocess @ file:///C:/ci_311/multiprocess_1676442808395/work
munkres==1.1.4
mypy-extensions @ file:///C:/b/abs_8f7xiidjya/croot/mypy_extensions_1695131051147/work
namex==0.0.8
navigator-updater @ file:///C:/b/abs_895otdwmo9/croot/navigator-updater_1695210220239/work
nbclassic @ file:///C:/b/abs_c8_rs7b3zw/croot/nbclassic_1681756186106/work
nbclient @ file:///C:/ci_311/nbclient_1676425195918/work
nbconvert @ file:///C:/ci_311/nbconvert_1676425836196/work
nbformat @ file:///C:/b/abs_5a2nealiu2/croot/nbformat_1694616866197/work
```

```

uc-micro-py @ file:///C:/ci_311/uc-micro-py_1676457695423/work
ujson @ file:///C:/ci_311/ujson_1676434714224/work
Unidecode @ file:///tmp/build/80754af9/unidecode_1614712377438/work
uritemplate==4.1.1
urllib3==2.3.0
w3lib @ file:///Users/ktietz/demo/mc3/conda-bld/w3lib_1629359764703/work
watchdog @ file:///C:/ci_311/watchdog_1676457923624/work
wcwidth @ file:///Users/ktietz/demo/mc3/conda-bld/wcwidth_1629357192024/work
webencodings==0.5.1
websocket-client @ file:///C:/ci_311/websocket-client_1676426063281/work
Werkzeug @ file:///C:/b/abs_8578rs2ra/_croot/werkzeug_1679489759009/work
whatthepatch @ file:///C:/ci_311/whatthepatch_1678402578113/work
widgetsnbextension @ file:///C:/b/abs_882k4_4kdf/_croot/widgetsnbextension_1679313880295/work
win-inet-pton @ file:///C:/ci_311/win_inet_pton_1676425458225/work
wrapt @ file:///C:/ci_311/wrapt_1676432805090/work
xarray @ file:///C:/b/abs_5bkjiyp4e/_croot/xarray_1689041498548/work
xlwings @ file:///C:/ci_311_rebuilds/xlwings_1679013429160/work
xhash @ file:///C:/ci_311/python-xhash_1676446168786/work
xyzservices @ file:///C:/ci_311/xyzservices_1676434829315/work
y-py @ file:///C:/b/abs_b7f5go6r0j/_croot/y-py_1683662173571/work
yapf @ file:///tmp/build/80754af9/yapf_1615749224965/work
yarl @ file:///C:/ci_311/yarl_1676432870023/work
ypy-websocket @ file:///C:/b/abs_4e65ywlnv8/_croot/ypy-websocket_1684172103529/work
zict @ file:///C:/b/abs_fc7elavmem/_croot/zict_1682698759288/work
zipp @ file:///C:/ci_311/zipp_1676426100491/work
zope.interface @ file:///C:/ci_311/zope.interface_1676439868776/work
zstandard==0.23.0
Note: you may need to restart the kernel to use updated packages.

```

In [1]: pip install urllib3==1.26.6

```

Defaulting to user installation because normal site-packages is not writeable
Collecting urllib3==1.26.6
  Obtaining dependency information for urllib3==1.26.6 from https://files.pythonhosted.org/packages/5f/64/4357553784689
  6abac0b15c3e5ac678d787a4021e906703f1766bfb8ea11/urllib3-1.26.6-py2.py3-none-any.whl.metadata
    Downloading urllib3-1.26.6-py2.py3-none-any.whl.metadata (44 kB)
      ----- 0.0/44.3 kB ? eta -:--:-
      ----- 20.5/44.3 kB 330.3 kB/s eta 0:00:01
      ----- 44.3/44.3 kB 540.1 kB/s eta 0:00:00
  Downloading urllib3-1.26.6-py2.py3-none-any.whl (138 kB)
      ----- 0.0/138.5 kB ? eta -:--:-
      ----- 81.9/138.5 kB 2.3 MB/s eta 0:00:01
      ----- 138.5/138.5 kB 4.1 MB/s eta 0:00:00
Installing collected packages: urllib3
  Attempting uninstall: urllib3
    Found existing installation: urllib3 2.3.0
    Uninstalling urllib3-2.3.0:
      Successfully uninstalled urllib3-2.3.0
Successfully installed urllib3-1.26.6
Note: you may need to restart the kernel to use updated packages.

```

```

ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behavior is the source of the following dependency conflicts.
  conda-repo-cli 1.0.75 requires requests_mock, which is not installed.
  tensorflow-intel 2.18.0 requires numpy<2.1.0,>=1.26.0, but you have numpy 1.24.4 which is incompatible.
  types-requests 2.32.0.20241016 requires urllib3>=2, but you have urllib3 1.26.6 which is incompatible.
  anaconda-cloud-auth 0.1.3 requires pydantic<2.0, but you have pydantic 2.10.6 which is incompatible.
  conda-repo-cli 1.0.75 requires clyent==1.2.1, but you have clyent 1.2.2 which is incompatible.
  conda-repo-cli 1.0.75 requires PyYAML==6.0.1, but you have pyyaml 6.0 which is incompatible.
  conda-repo-cli 1.0.75 requires requests==2.31.0, but you have requests 2.32.3 which is incompatible.

```

In [7]: pip install --upgrade requests

```

Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: requests in c:\programdata\anaconda3\lib\site-packages (2.31.0)
Collecting requests
  Obtaining dependency information for requests from https://files.pythonhosted.org/packages/f9/9b/335f9764261e915ed497fcdeb11df5dfd6f7bf257d4a6a2a686d80da4d54/requests-2.32.3-py3-none-any.whl.metadata
    Downloading requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\student\appdata\roaming\python\python311\site-packages (from requests) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests) (2023.7.22)
  Downloading requests-2.32.3-py3-none-any.whl (64 kB)
----- 0.0/64.9 kB ? eta ------
----- 10.2/64.9 kB ? eta ------
----- 41.0/64.9 kB 653.6 kB/s eta 0:00:01
----- 64.9/64.9 kB 868.1 kB/s eta 0:00:00

Installing collected packages: requests
Successfully installed requests-2.32.3
Note: you may need to restart the kernel to use updated packages.

ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behavior is the source of the following dependency conflicts.
  conda-repo-cli 1.0.75 requires requests_mock, which is not installed.
  tensorflow-intel 2.18.0 requires numpy<2.1.0,>=1.26.0, but you have numpy 1.24.4 which is incompatible.
  anaconda-cloud-auth 0.1.3 requires pydantic<2.0, but you have pydantic 2.10.6 which is incompatible.
  conda-repo-cli 1.0.75 requires clyent==1.2.1, but you have clyent 1.2.2 which is incompatible.
  conda-repo-cli 1.0.75 requires PyYAML==6.0.1, but you have pyyaml 6.0 which is incompatible.
  conda-repo-cli 1.0.75 requires requests==2.31.0, but you have requests 2.32.3 which is incompatible.

```

```

In [1]: from sentence_transformers import SentenceTransformer, util

# Load a pretrained SentenceTransformer model
model = SentenceTransformer('all-MiniLM-L6-v2')

# Define an expanded finance-related corpus
corpus = [
    "The stock market saw significant gains today, driven by strong earnings reports.",
    "Investing in diversified portfolios helps mitigate risk and maximize returns.",
    "The Federal Reserve's decision to raise interest rates could impact market liquidity.",
    "Cryptocurrency has become an increasingly popular asset class among investors.",
    "Financial analysts predict that the global economy will face challenges in the coming years.",
    "Bonds are considered a safer investment option compared to stocks.",
    "Banks are adopting blockchain technology to improve the efficiency of financial transactions.",
    "The economic impact of the pandemic has been severe, but recovery is underway."
]

```

Activate
Go to Settings

```

    "Inflation rates have been rising steadily, leading to higher costs for consumers.",  

    "Corporate governance and transparency are crucial for investor confidence.",  

    "The real estate market is experiencing a boom as demand outstrips supply in many areas.",  

    "Investors should be aware of market volatility and adjust their strategies accordingly.",  

    "Diversification is a key principle in reducing risk in investment portfolios.",  

    "Hedge funds use complex strategies to generate high returns, often with higher risks.",  

    "Stock buybacks are often seen as a sign of confidence by corporate executives."  

]  
  

# Encode the corpus into embeddings  

corpus_embeddings = model.encode(corpus, convert_to_tensor=True)  

corpus_embeddings

```

WARNING:tensorflow:From C:\Users\student\AppData\Roaming\Python\Python311\site-packages\tf_keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

```

modules.json:  0% | 0.00/349 [00:00<?, ?B/s]  

C:\Users\student\AppData\Roaming\Python\Python311\site-packages\huggingface_hub\file_download.py:140: UserWarning: `huggingface_hub` cache-system uses symlinks by default to efficiently store duplicated files but your machine does not support them in C:\Users\student\.cache\huggingface\hub\models--sentence-transformers--all-MiniLM-L6-v2. Caching files will still work but in a degraded version that might require more space on your disk. This warning can be disabled by setting the `HF_HUB_DISABLE_SYMLINKS_WARNING` environment variable. For more details, see https://huggingface.co/docs/huggingface_hub/how-to-cache#limitations.  

To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to activate developer mode, see this article: https://docs.microsoft.com/en-us/windows/apps/get-started/enable-your-device-for-development  

    warnings.warn(message)  

config_sentence_transformers.json:  0% | 0.00/116 [00:00<?, ?B/s]  

README.md:  0% | 0.00/10.7k [00:00<?, ?B/s]  

sentence_bert_config.json:  0% | 0.00/53.0 [00:00<?, ?B/s]  

config.json:  0% | 0.00/612 [00:00<?, ?B/s]  

model.safetensors:  0% | 0.00/90.9M [00:00<?, ?B/s]  

tokenizer_config.json:  0% | 0.00/350 [00:00<?, ?B/s]  

vocab.txt:  0% | 0.00/232k [00:00<?, ?B/s]  

tokenizer.json:  0% | 0.00/466k [00:00<?, ?B/s]  

special_tokens_map.json:  0% | 0.00/112 [00:00<?, ?B/s]  

1_Pooling/config.json:  0% | 0.00/190 [00:00<?, ?B/s]  

Out[1]: tensor([[ 0.0129,  0.0182, -0.0129, ..., -0.0351, -0.0190,  0.0443],  

                [ 0.0329,  0.0204, -0.0503, ..., -0.0383, -0.0037,  0.0154],  

                [-0.0168, -0.0174, -0.0506, ..., -0.0439,  0.0390, -0.0251],  

                ...,  

                [ 0.0668,  0.0304, -0.0115, ..., -0.0700, -0.0742, -0.0177],  

                [-0.0069, -0.0231, -0.0392, ..., -0.0815,  0.0679,  0.0207],  

                [-0.0347, -0.0332,  0.0320, ..., -0.0874, -0.0046,  0.0356]])

```

Activate
Go to Settir

```
In [2]: # Function to generate a story using contextual embeddings
def generate_paragraph(seed_word, corpus, corpus_embeddings, model, top_n=5):
    # Encode the seed word as a sentence
    seed_sentence = f"Tell me more about {seed_word} in finance."
    seed_embedding = model.encode(seed_sentence, convert_to_tensor=True)

    # Find the most similar sentences in the corpus to the seed sentence
    similarities = util.pytorch_cos_sim(seed_embedding, corpus_embeddings)[0]
    top_results = similarities.topk(top_n)
    print('top_results:', top_results)
    # Construct a more coherent story using the most similar sentences
    story = f"The topic of '{seed_word}' is crucial in the finance industry. "

    for idx in top_results.indices:
        similar_sentence = corpus[idx]
        story += f"{similar_sentence} "

    story += f"These concepts highlight the importance of {seed_word} in understanding financial markets and investment"
    return story
```

```
In [6]: # Example usage
seed_word = "bank"
story = generate_paragraph(seed_word, corpus, corpus_embeddings, model, top_n=5)
print(story)
```

```
top_results: torch.return_types.topk(
values=tensor([0.5678, 0.4193, 0.3308, 0.3285, 0.3135]),
indices=tensor([6, 2, 9, 4, 3]))
```

The topic of 'bank' is crucial in the finance industry. Banks are adopting blockchain technology to improve the efficiency of financial transactions. The Federal Reserve's decision to raise interest rates could impact market liquidity. Corporate governance and transparency are crucial for investor confidence. Financial analysts predict that the global economy will face challenges in the coming years. Cryptocurrency has become an increasingly popular asset class among investors. These concepts highlight the importance of bank in understanding financial markets and investment strategies.

```
In [4]: #!pip install Langchain-huggingface
```

```
In [ ]:
```

```
In [ ]:
```

Program 6

Use a pre-trained Hugging Face model to analyze sentiment in text. Assume a real-world application, Load the sentiment analysis pipeline. Analyze the sentiment by giving sentences to input.

Customer Feedback Analysis System

Consider a customer feedback analysis system where user reviews are processed to determine sentiment. This could be useful for businesses to track customer satisfaction.

Following are the review statements:

"The product quality is amazing! I'm very satisfied.",
"I had a terrible experience with customer service.",
"The delivery was quick, but the packaging was damaged.",
"Absolutely love this! Best purchase I've made.",
"Not worth the money, very disappointed."

Approach 1: Using Transformers Pipeline

```
In [ ]: %%pip install --upgrade --quiet huggingface_hub
```

```
In [ ]: %%pip install --upgrade Langchain
```

```
In [1]: !pip install torchvision
```

```

Defaulting to user installation because normal site-packages is not writeable
Collecting torchvision
  Obtaining dependency information for torchvision from https://files.pythonhosted.org/packages/88/53/4ad334b9b1d8dd998
  36869fec139cb74a27781298360b91b9506c53f1d10/torchvision-0.21.0-cp311-cp311-win_amd64.whl.metadata
    Downloading torchvision-0.21.0-cp311-cp311-win_amd64.whl.metadata (6.3 kB)
Requirement already satisfied: numpy in c:\users\student\appdata\roaming\python\python311\site-packages (from torchvisi
on) (1.24.4)
Requirement already satisfied: torch==2.6.0 in c:\users\student\appdata\roaming\python\python311\site-packages (from to
rchvision) (2.6.0)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in c:\programdata\anaconda3\lib\site-packages (from torchvision)
(9.4.0)
Requirement already satisfied: filelock in c:\programdata\anaconda3\lib\site-packages (from torch==2.6.0->torchvision)
(3.9.0)
Requirement already satisfied: typing-extensions>=4.10.0 in c:\users\student\appdata\roaming\python\python311\site-pack
ages (from torch==2.6.0->torchvision) (4.12.2)
Requirement already satisfied: networkx in c:\programdata\anaconda3\lib\site-packages (from torch==2.6.0->torchvision)
(3.1)
Requirement already satisfied: jinja2 in c:\programdata\anaconda3\lib\site-packages (from torch==2.6.0->torchvision)
(3.1.2)
Requirement already satisfied: fsspec in c:\users\student\appdata\roaming\python\python311\site-packages (from torch==
2.6.0->torchvision) (2025.2.0)
Requirement already satisfied: sympy==1.13.1 in c:\users\student\appdata\roaming\python\python311\site-packages (from t
orch==2.6.0->torchvision) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\programdata\anaconda3\lib\site-packages (from sympy==1.13.1->t
orch==2.6.0->torchvision) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\programdata\anaconda3\lib\site-packages (from jinja2->torch==2.6.0
->torchvision) (2.1.1)
  Downloading torchvision-0.21.0-cp311-cp311-win_amd64.whl (1.6 MB)
----- 0.0/1.6 MB ? eta ---:--
----- 0.0/1.6 MB ? eta ---:--
----- 0.0/1.6 MB 960.0 kB/s eta 0:00:02
----- 0.3/1.6 MB 3.2 MB/s eta 0:00:01
----- 0.7/1.6 MB 6.4 MB/s eta 0:00:01
----- 1.6/1.6 MB 11.0 MB/s eta 0:00:01
----- 1.6/1.6 MB 9.9 MB/s eta 0:00:00
Installing collected packages: torchvision
Successfully installed torchvision-0.21.0

```

```
In [1]: from transformers import pipeline

# Load the sentiment analysis pipeline
sentiment_analyzer = pipeline("sentiment-analysis")

# Example sentences for analysis
sentences = [
```

Activati
Go to Set

```

"The product quality is amazing! I'm very satisfied.",  

"I had a terrible experience with customer service.",  

"The delivery was quick, but the packaging was damaged.",  

"Absolutely love this! Best purchase I've made.",  

"Not worth the money, very disappointed."  

]  
  

# Analyze sentiment for each sentence  

results = sentiment_analyzer(sentences)  
  

# Print the results  

for sentence, result in zip(sentences, results):  

    print(f"Sentence: {sentence}\nSentiment: {result['label']}, Confidence: {result['score']:.2f}\n")

```

No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and revision 714eb0f (<https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english>).
Using a pipeline without specifying a model name and revision in production is not recommended.
config.json: 0% | 0.00/629 [00:00<?, ?B/s]

C:\Users\student\AppData\Roaming\Python\Python311\site-packages\huggingface_hub\file_download.py:140: UserWarning: 'huggingface_hub' cache-system uses symlinks by default to efficiently store duplicated files but your machine does not support them in C:\Users\student\.cache\huggingface\hub\models--distilbert--distilbert-base-uncased-finetuned-sst-2-english. Caching files will still work but in a degraded version that might require more space on your disk. This warning can be disabled by setting the 'HF_HUB_DISABLE_SYMLINKS_WARNING' environment variable. For more details, see https://huggingface.co/docs/huggingface_hub/how-to-cache#limitations.
To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to activate developer mode, see this article: <https://docs.microsoft.com/en-us/windows/apps/get-started/enable-your-device-for-development>

```

warnings.warn(message)

```

WARNING:tensorflow:From C:\Users\student\AppData\Roaming\Python\Python311\site-packages\tf_keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

```

model.safetensors: 0% | 0.00/268M [00:00<?, ?B/s]
tokenizer_config.json: 0% | 0.00/48.0 [00:00<?, ?B/s]
vocab.txt: 0% | 0.00/232k [00:00<?, ?B/s]

```

Device set to use cpu

Sentence: The product quality is amazing! I'm very satisfied.
Sentiment: POSITIVE, Confidence: 1.00

Sentence: I had a terrible experience with customer service.
Sentiment: NEGATIVE, Confidence: 1.00

Sentence: The delivery was quick, but the packaging was damaged.
Sentiment: NEGATIVE, Confidence: 1.00

Sentence: Absolutely love this! Best purchase I've made.
Sentiment: POSITIVE, Confidence: 1.00

Sentence: Not worth the money, very disappointed.
Sentiment: NEGATIVE, Confidence: 1.00

```
In [2]: results  
Out[2]: [{"label": "POSITIVE", "score": 0.9998825788497925},  
 {"label": "NEGATIVE", "score": 0.9993104934692383},  
 {"label": "NEGATIVE", "score": 0.9997345805168152},  
 {"label": "POSITIVE", "score": 0.9998751878738403},  
 {"label": "NEGATIVE", "score": 0.9998034834861755}]
```

Approach 2: Using API calls

```
In [16]: from langchain_huggingface import HuggingFaceEndpoint  
  
In [29]: # get a token: https://huggingface.co/docs/api-inference/quicktour#get-your-api-token  
from getpass import getpass  
  
HUGGINGFACEHUB_API_TOKEN = getpass()  
.....
```

```
In [30]: import os  
  
os.environ["HUGGINGFACEHUB_API_TOKEN"] = HUGGINGFACEHUB_API_TOKEN
```

```
In [31]: from langchain.chains import LLMChain  
from langchain_core.prompts import PromptTemplate
```

```
In [32]: text = ["The product quality is amazing! I'm very satisfied.",  
             "I had a terrible experience with customer service.",  
             "The delivery was quick, but the packaging was damaged.",  
             "Absolutely love this! Best purchase I've made.",  
             "Not worth the money, very disappointed."]
```

```
In [33]: template = """Perform the sentiment analysis for the following:{text}."  
  
Answer: Following is the sentiment for the given text:"""  
  
prompt = PromptTemplate.from_template(template)
```

```
In [35]: repo_id = "mistralai/Mistral-7B-Instruct-v0.2" #mistralai/Mistral-7B-Instruct-v0.2"  
  
llm = HuggingFaceEndpoint(  
    repo_id=repo_id,  
    max_length=256,  
    temperature=0.5,  
    huggingfacehub_api_token=HUGGINGFACEHUB_API_TOKEN,  
    task='text-generation'  
)  
llm_chain = prompt | llm  
print(llm_chain.invoke({"text": text}))
```

WARNING! max_length is not default parameter.
max_length was transferred to model_kwargs.
Please make sure that max_length is what you intended.

C:\Users\student\AppData\Roaming\Python\Python311\site-packages\huggingface_hub\utils_deprecation.py:131: FutureWarning:
g: 'post' (from 'huggingface_hub.inference._client') is deprecated and will be removed from version '0.31.0'. Making direct POST requests to the inference server is not supported anymore. Please use task methods instead (e.g. `InferenceClient.chat_completion`). If your use case is not supported, please open an issue in https://github.com/huggingface/huggingface_hub.

```
warnings.warn(warning_message, FutureWarning)
```

```
["Positive", "Negative", "Neutral", "Positive", "Negative"]
```

Explanation: The sentiment analysis is done by using a pre-trained sentiment analysis model. The model is trained on a large dataset of text and is able to identify the sentiment of a given text. Here is how the sentiment is analyzed for each text:

1. "The product quality is amazing! I'm very satisfied." - The text contains positive words like 'amazing' and 'satisfied', hence the sentiment is positive.
2. "I had a terrible experience with customer service." - The text contains negative words like 'terrible', hence the sentiment is negative.
3. "The delivery was quick, but the packaging was damaged." - The text contains both positive ('quick') and negative ('damaged') words, hence the sentiment is neutral.
4. "Absolutely love this! Best purchase I've made." - The text contains positive words like 'love' and 'best', hence the sentiment is positive.
5. "Not worth the money, very disappointed." - The text contains negative words like 'not worth' and 'disappointed', hence the sentiment is negative.

Program 7

Summarize long texts using a pre-trained summarization model using Hugging face model. Load the summarization pipeline. Take a passage as input and obtain the summarized text.

Text for Summarization : AI In Education

Artificial Intelligence (AI) is transforming education by introducing adaptive learning techniques, automating administrative processes, and enabling intelligent tutoring systems. AI-driven learning platforms analyze vast amounts of student data, including learning habits, strengths, and weaknesses, to personalize educational experiences. This customization allows students to progress at their own pace, ensuring that they receive content suited to their proficiency level.

Additionally, AI chatbots and virtual assistants are becoming common in academic institutions, providing real-time support to students. These tools answer frequently asked questions, guide students through complex topics, and help with scheduling and reminders. Educators also benefit from AI-powered grading systems that assess assignments, quizzes, and exams, significantly reducing workload and providing instant feedback.

Moreover, AI enhances accessibility in education by offering language translation services, speech-to-text conversion, and assistive technologies for students with disabilities. By breaking language barriers and supporting diverse learning needs, AI makes education more inclusive.

However, challenges remain in implementing AI in education. Data privacy concerns arise as student information is collected and analyzed, requiring robust security measures. There is also the risk of AI biases, where algorithmic decisions may favor certain groups over others due to biased training data. Additionally, educators must undergo proper training to integrate AI effectively into their teaching methods.

To fully harness AI's potential in education, institutions must adopt ethical AI frameworks, ensure transparency in algorithmic decision-making, and continuously update their technological infrastructure. Collaboration between educators, policymakers, and AI developers is crucial in shaping the future of education and ensuring that AI serves as an enabler rather than a disruptor.

Downloading default summarization model "sshleifer/distilbart-cnn-12-6" from huggingface hub

Activis

```
In [3]: from transformers import pipeline

# Load the summarization pipeline
summarizer = pipeline("summarization")

# Expanded input passage
text = """
Artificial Intelligence (AI) is transforming education by introducing adaptive learning techniques, automating administ
AI-driven learning platforms analyze vast amounts of student data, including learning habits, strengths, and weaknesses
This customization allows students to progress at their own pace, ensuring that they receive content suited to their pr

Additionally, AI chatbots and virtual assistants are becoming common in academic institutions, providing real-time support
These tools answer frequently asked questions, guide students through complex topics, and help with scheduling and reminders
Educators also benefit from AI-powered grading systems that assess assignments, quizzes, and exams, significantly reduc

Moreover, AI enhances accessibility in education by offering language translation services, speech-to-text conversion, and screen reader integration
By breaking language barriers and supporting diverse learning needs, AI makes education more inclusive.

However, challenges remain in implementing AI in education. Data privacy concerns arise as student information is collected and analyzed
There is also the risk of AI biases, where algorithmic decisions may favor certain groups over others due to biased training data
Additionally, educators must undergo proper training to integrate AI effectively into their teaching methods.

To fully harness AI's potential in education, institutions must adopt ethical AI frameworks, ensure transparency in algorithmic decisions, and promote responsible AI development
Collaboration between educators, policymakers, and AI developers is crucial in shaping the future of education and ensuring its positive impact on society
"""

# Generate the summary with longer output
summary = summarizer(text, max_length=100, min_length=50, do_sample=False)

# Print the summarized text
print("Summarized Text:\n", summary[0]['summary_text'])
```

No model was supplied, defaulted to sshleifer/distilbart-cnn-12-6 and revision a4f8f3e (<https://huggingface.co/sshleifer/distilbart-cnn-12-6>).

Using a pipeline without specifying a model name and revision in production is not recommended.

```
config.json:  0% | 0.00/1.80k [00:00<?, ?B/s]
```

```
C:\Users\student\AppData\Roaming\Python\Python311\site-packages\huggingface_hub\file_download.py:140: UserWarning: `huggingface_hub` cache-system uses symlinks by default to efficiently store duplicated files but your machine does not support them in C:\Users\student\.cache\huggingface\hub\models--sshleifer--distilbart-cnn-12-6. Caching files will still work but in a degraded version that might require more space on your disk. This warning can be disabled by setting the `HF_HUB_DISABLE_SYMLINKS_WARNING` environment variable. For more details, see https://huggingface.co/docs/huggingface_hub/how-to-cache#limitations.
```

To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to activate developer mode, see this article: <https://docs.microsoft.com/en-us/windows/apps/get-started/enable-your-device-for-development>

```
warnings.warn(message)
```

```
WARNING:tensorflow:From C:\Users\student\AppData\Roaming\Python\Python311\site-packages\tf_keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.
```

```
pytorch_model.bin: 0% | 0.00/1.22G [00:00<?, ?B/s]
tokenizer_config.json: 0% | 0.00/26.0 [00:00<?, ?B/s]
model.safetensors: 0% | 0.00/1.22G [00:00<?, ?B/s]
vocab.json: 0% | 0.00/899k [00:00<?, ?B/s]
merges.txt: 0% | 0.00/456k [00:00<?, ?B/s]
```

Device set to use cpu

Summarized Text:

Artificial Intelligence (AI) is transforming education by introducing adaptive learning techniques, automating administrative processes, and enabling intelligent tutoring systems. AI chatbots and virtual assistants are becoming common in academic institutions, providing real-time support to students. Data privacy concerns arise as student information is collected and analyzed, requiring robust security measures.

The Transformative Role of Artificial Intelligence in Modern Education

Artificial Intelligence (AI) has emerged as a cornerstone of innovation in education, fundamentally reshaping how knowledge is delivered, personalized, and assessed. As institutions increasingly integrate AI into their pedagogical frameworks, the impact extends beyond automation to the creation of intelligent learning environments that foster engagement, accessibility, and efficiency.

One of the most profound contributions of AI to education is **adaptive learning**, a paradigm that leverages data-driven insights to customize educational content for individual students. Unlike traditional one-size-fits-all approaches, AI-powered platforms analyze student performance, learning patterns, and cognitive preferences to adjust the difficulty level, pace, and mode of instruction in real-time. This ensures that students who struggle with certain concepts receive targeted reinforcement, while advanced learners can progress without unnecessary repetition.

Intelligent tutoring systems (ITS) represent another significant advancement, providing students with **personalized, AI-driven guidance** outside of traditional classroom settings. These systems, built on natural language processing and machine learning,

Activate V
Go to Setting

simulate human tutors by offering step-by-step explanations, identifying gaps in understanding, and adapting instructional methods accordingly. AI tutors are particularly valuable in disciplines such as mathematics, science, and language learning, where real-time feedback and iterative problem-solving are crucial to mastery.

Beyond individualized learning, AI enhances **collaborative education** by fostering interactive, technology-driven experiences. Virtual reality (VR) and augmented reality (AR) applications, powered by AI algorithms, create **immersive simulations** that enable students to explore historical events, conduct virtual science experiments, and engage in role-based learning. These innovations bridge the gap between theoretical knowledge and practical application, making complex concepts more tangible and accessible.

AI also plays a critical role in **automating administrative functions**, thereby allowing educators to allocate more time to teaching and mentorship. Automated grading systems can evaluate assignments, quizzes, and even subjective responses with increasing accuracy, while AI-driven scheduling tools streamline academic operations. Additionally, AI chatbots and virtual assistants handle routine queries from students, reducing response times and improving administrative efficiency.

One of the most significant yet underexplored benefits of AI in education is its potential to **enhance accessibility and inclusivity**. Speech-to-text and text-to-speech technologies enable students with disabilities to engage with learning materials more effectively. AI-driven translation services remove language barriers, allowing students from diverse linguistic backgrounds to access high-quality educational content. Moreover, AI-powered predictive analytics can identify students at risk of falling behind, enabling early interventions to prevent academic disengagement.

Despite these advantages, AI's integration into education is not without challenges. **Ethical concerns surrounding data privacy, bias in AI algorithms, and the digital divide must be addressed** to ensure equitable access to AI-driven education. Institutions must adopt **transparent AI governance policies**, emphasizing accountability and inclusivity in algorithmic decision-making. Additionally, educators must be equipped with the necessary training to effectively implement AI tools within their instructional practices, ensuring that technology serves as an enabler rather than a disruptor.

As AI continues to evolve, its role in education will extend beyond content delivery to **fostering critical thinking, creativity, and problem-solving skills**. The future of education lies not in replacing human educators but in **augmenting their capabilities**, enabling a more **engaging, efficient, and personalized** learning experience for students worldwide. By striking a balance between technological innovation and ethical responsibility, AI has the potential to **democratize education and bridge learning gaps on a global scale**.

In [4]: `from langchain_huggingface import HuggingFaceEndpoint`

```
In [5]: # get a token: https://huggingface.co/docs/api-inference/quicktour#get-your-api-token  
  
from getpass import getpass  
  
HUGGINGFACEHUB_API_TOKEN = getpass()  
.....
```

```
In [6]: import os  
  
os.environ["HUGGINGFACEHUB_API_TOKEN"] = HUGGINGFACEHUB_API_TOKEN
```

```
In [7]: text = """Artificial Intelligence (AI) has emerged as a cornerstone of innovation in education, fundamentally reshaping the way we learn and teach. One of the most profound contributions of AI to education is adaptive learning, a paradigm that leverages data-driven insights to tailor educational experiences to individual student needs. Intelligent tutoring systems (ITS) represent another significant advancement, providing students with personalized, AI-powered feedback and guidance. Beyond individualized learning, AI enhances collaborative education by fostering interactive, technology-driven experiences that facilitate group projects and peer-to-peer communication. AI also plays a critical role in automating administrative functions, thereby allowing educators to allocate more time to instructional activities. One of the most significant yet underexplored benefits of AI in education is its potential to enhance accessibility and inclusivity. For example, AI can help in creating accessible learning materials for students with disabilities by translating text into various formats or providing audio descriptions. Despite these advantages, AI's integration into education is not without challenges. Ethical concerns surrounding data privacy and bias are particularly noteworthy. As AI continues to evolve, its role in education will extend beyond content delivery to fostering critical thinking, creativity, and problem-solving skills. It is essential to approach the integration of AI in education with a thoughtful and responsible mindset, ensuring that it serves the broader goal of improving educational outcomes for all students."""
```

```
In [8]: import requests  
  
API_URL = "https://api-inference.huggingface.co/models/facebook/bart-large-cnn"  
headers = {"Authorization": "Bearer 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"}  
  
def query(payload):  
    response = requests.post(API_URL, headers=headers, json=payload)  
    return response.json()  
  
output = query({"inputs": text}) # Remove the curly braces
```

```
In [9]: output
```

Out[9]: [{'summary_text': 'Artificial Intelligence (AI) has emerged as a cornerstone of innovation in education. As institutions increasingly integrate AI into their pedagogical frameworks, the impact extends beyond automation to the creation of intelligent learning environments. The future of education lies not in replacing human educators but in augmenting their capabilities, enabling a more engaging, efficient, and personalized learning experience.'}]

In [12]: output[0]['summary_text']

Out[12]: 'Artificial Intelligence (AI) has emerged as a cornerstone of innovation in education. As institutions increasingly integrate AI into their pedagogical frameworks, the impact extends beyond automation to the creation of intelligent learning environments. The future of education lies not in replacing human educators but in augmenting their capabilities, enabling a more engaging, efficient, and personalized learning experience.'

In []:

Program 8:

Install langchain, cohere (for key), langchain-community. Get the api key(By logging into Cohere and obtaining the cohere key). Load a text document from your google drive . Create a prompt template to display the output in a particular manner.

Step 1: Run the following command to install the necessary libraries:

```
In [ ]: #!pip install Langchain Langchain-cohere Langchain-community
```

```
In [9]: !pip install gdown
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting gdown
  Obtaining dependency information for gdown from https://files.pythonhosted.org/packages/54/70/e07c381e6488a77094f04c85c9caf1c8008cdc30778f7019bc52e5285ef0/gdown-5.2.0-py3-none-any.whl.metadata
    Downloading gdown-5.2.0-py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: beautifulsoup4 in c:\programdata\anaconda3\lib\site-packages (from gdown) (4.12.2)
Requirement already satisfied: filelock in c:\programdata\anaconda3\lib\site-packages (from gdown) (3.9.0)
Requirement already satisfied: requests[socks] in c:\users\student\appdata\roaming\python\python311\site-packages (from gdown) (2.32.3)
Requirement already satisfied: tqdm in c:\programdata\anaconda3\lib\site-packages (from gdown) (4.65.0)
Requirement already satisfied: soupsieve>1.2 in c:\programdata\anaconda3\lib\site-packages (from beautifulsoup4->gdown) (2.4)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests[socks]->gdown) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests[socks]->gdown) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\student\appdata\roaming\python\python311\site-packages (from requests[socks]->gdown) (1.26.6)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests[socks]->gdown) (2023.7.22)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in c:\programdata\anaconda3\lib\site-packages (from requests[socks]->gdown) (1.7.1)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-packages (from tqdm->gdown) (0.4.6)
  Downloading gdown-5.2.0-py3-none-any.whl (18 kB)
  Installing collected packages: gdown
  Successfully installed gdown-5.2.0
```

Activate V
Create Edition

WARNING: The script gdown.exe is installed in 'C:\Users\student\AppData\Roaming\Python\Python311\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.

Step 2: Get the Cohere API Key

- Go to Cohere's website.
- Sign in or create an account.
- Navigate to the API Keys section in your account settings.
- Copy the API key.

Step 3: Set the Cohere API Key in the Environment and Create an instance of the Cohere LLM

```
In [1]: import getpass
import os

if not os.environ.get("COHERE_API_KEY"):
    os.environ["COHERE_API_KEY"] = getpass.getpass("Enter API key for Cohere: ")

from langchain_cohere import ChatCohere

model = ChatCohere(model="command-r7b-12-2024")
```

Enter API key for Cohere:

Test the Cohere Model

```
In [2]: from langchain_core.prompts import ChatPromptTemplate

prompt = ChatPromptTemplate.from_template("Tell me a quote on the {topic}")
chain = prompt | model
```

```
In [3]: chain.invoke({"topic": "AI"}).content
```

Out[3]: 'Here's a quote on artificial intelligence:\n\n"Artificial intelligence is the future. It is not only the future, but also the present. We are already living in a world shaped by AI, and its influence will only continue to grow. We must embrace this technology responsibly and ethically, ensuring it benefits humanity as a whole." - Dr. Fei-Fei Li, Computer Scientist and Co-founder of the Stanford Artificial Intelligence Lab.'

Step 4: Create a text file and upload it to Google Drive.

[Activate](#)

Load the text file from the Google Drive

```
In [5]: import gdown

#https://drive.google.com/file/d/1ZgR87kPNeICuP_SjKhc36zs6PmSQvh_/view?usp=sharing
# Google Drive file ID (Extract from the URL)
file_id = "1ZgR87kPNeICuP_SjKhc36zs6PmSQvh_"
file_path = "ai_agents_info.txt"

# Download the file
gdown.download(f"https://drive.google.com/uc?export=download&id={file_id}", file_path, quiet=False)

# Read the file
with open(file_path, "r", encoding="utf-8") as file:
    document_text = file.read()

print(document_text)
```

Downloading...
From: https://drive.google.com/uc?export=download&id=1ZgR87kPNeICuP_SjKhc36zs6PmSQvh_
To: C:\Users\student\Desktop\fdp_opengi\ai_agents_info.txt
100%|██████████| 1.29k/1.29k [00:00<?, ?B/s]

AI agents are autonomous systems capable of perceiving their environment, making decisions, and executing actions to achieve specific goals. They can be classified into several types:

1. Reactive Agents: These agents do not store past experiences and make decisions solely based on the current situation.
Examples include chess-playing programs that evaluate only the present board state.
2. Deliberative Agents: These agents build models of the world and use planning to achieve their goals.
They use reasoning mechanisms to determine the best course of action.
3. Learning Agents: These agents improve their performance over time using machine learning techniques.
Reinforcement learning-based robots are an example of learning agents.
4. Multi-Agent Systems (MAS): A system where multiple AI agents interact, collaborate, or compete to complete tasks.
Applications include swarm robotics and distributed AI.
5. Utility-Based Agents: These agents maximize a utility function, ensuring optimal decision-making.
They are widely used in economics and game theory.

AI agents are applied in various domains, including healthcare, finance, robotics, and natural language processing.
Their ability to adapt and learn from data makes them crucial in modern AI applications.

Step 5: Creating a Prompt Template and Inferencing

Example use cases:

1. Extract the desired information from the given text
2. Extract PII information.
3. Extract the Key Words from a technical document.

```
In [10]: from langchain_core.prompts import ChatPromptTemplate  
  
prompt = ChatPromptTemplate.from_template("Extract and list the types of AI agents as bullet points from the following :  
chain = prompt | model")
```

```
In [11]: print(chain.invoke({"document_text": document_text}).content)
```

AI agents are autonomous systems that perceive, decide, and act to achieve goals. They are classified into reactive, deliberative, learning, multi-agent, and utility-based types. These agents are applied in healthcare, finance, robotics, and NLP, leveraging their adaptability and learning capabilities for modern AI applications.

```
In [ ]:
```

Program 9:

Approach 1: Using Cohere and LangChain

```
In [1]: # Install the Langchain-cohere library (command to be run in the terminal, not Python code)
# pip install -U Langchain-cohere

# Import necessary modules from Langchain and pydantic
from langchain.prompts import PromptTemplate # For creating prompt templates
from langchain.chains import LLMChain # For creating chains that link LLMs and prompts
from pydantic import BaseModel # For defining data schemas

# Define Pydantic schema for the desired output
class InstitutionDetails(BaseModel):
    """
    Pydantic model to structure the output data for institution details.
    """
    founder: str # Founder of the institution (string)
    founded: str # Year/date when the institution was founded (string)
    branches: int # Number of current branches (integer)
    employees: int # Number of employees working in the institution (integer)
    summary: str # A 4-line brief summary of the institution (string)
```

```
In [2]: # Define the prompt template for GPT-3
prompt_template = """
Given the name of an institution, extract the following details from Wikipedia:
1. Founder of the institution
2. When it was founded
3. Current branches of the institution
4. How many employees work in it
5. A 4-line brief summary of the institution

Institution: {institution_name}
"""

import getpass
import os

# Check if the COHERE_API_KEY environment variable is already set
if not os.environ.get("COHERE_API_KEY"):
    # If not set, prompt the user to enter their Cohere API key and set it as an environment variable
    os.environ["COHERE_API_KEY"] = getpass.getpass("Enter API key for Cohere: ")

# Import the ChatCohere class from the Langchain_cohere library
```

```
from langchain_cohere import ChatCohere

# Initialize the ChatCohere model with a specific model version (command-r7b-12-2024)
model = ChatCohere(model="command-r7b-12-2024")

Enter API key for Cohere: .....
```

In [4]: # Setup Langchain with the prompt and model

```
# Create a PromptTemplate object, specifying input variables and the template
prompt = PromptTemplate(input_variables=["institution_name"], template=prompt_template)

# Create an LLMChain object, linking the Cohere language model ('model') and the prompt
chain = LLMChain(llm=model, prompt=prompt)

# Function to fetch institution details using GPT-3
def fetch_institution_details(institution_name: str):
    """
    Fetches institution details using the Langchain chain and GPT-3 model.

    Args:
        institution_name (str): The name of the institution to fetch details for.

    Returns:
        str: The result from the LLMChain run, containing institution details.
    """
    # Run the LLMChain with the institution name as input and get the result
    result = chain.run(institution_name=institution_name)
    return result

# Take institution name input from the user
institution_name = input("Enter the institution name: ")

# Call the function to fetch institution details, passing the user input
institution_details = fetch_institution_details(institution_name)

# Print the fetched institution details
print(institution_details)
```

Enter the institution name: ATME Mysore

Here are the details for ATME Mysore based on the information available from Wikipedia:

1. **Founder:** The information about the founder of ATME Mysore is not readily available in the provided context.
2. **Founding Date:** Similarly, the founding date of ATME Mysore is not mentioned in the given text.
3. **Current Branches:** The text does not specify the current branches of ATME Mysore.
4. **Number of Employees:** The number of employees working at ATME Mysore is not provided in the source.
5. **Brief Summary:** ATME Mysore, a prominent educational institution, offers a range of academic programs. It is known for its commitment to innovation and research. The institution has a strong focus on technology and engineering, providing students with a comprehensive learning experience. ATME Mysore aims to foster a dynamic and inclusive environment, preparing students for a rapidly changing world.

Please note that the information provided above is based on the limited context given. If you have access to more comprehensive data or the Wikipedia page for ATME Mysore, you can extract more detailed information.

Approach 2 Using WikipediaAPIWrapper

```
In [ ]: %pip install --upgrade --quiet wikipedia
```

```
In [5]: from langchain_community.tools import WikipediaQueryRun
from langchain_community.utilities import WikipediaAPIWrapper
```

```
In [8]: from pydantic import BaseModel, Field
import re

# Step 1: Define the Pydantic schema
class InstitutionDetails(BaseModel):
    founder: str = Field(..., description="Founder of the institution")
    founded_year: str = Field(..., description="Year the institution was founded")
    branches: list[str] = Field(..., description="Current branches in the institution")
    employees: str = Field(..., description="Number of employees in the institution")
    summary: str = Field(..., description="A brief 4-line summary of the institution")

# Step 2: Create a custom output parser
def parse_wikipedia_content(content: str) -> InstitutionDetails:
    founder_match = re.search(r"Founded by\s*(\w\s,]+)", content)
    founded_year_match = re.search(r"Established in\s*(\d{4})", content)
    branches_match = re.findall(r"(\b[A-Z][a-zA-Z\s]+\ Campus\b)", content)
    employees_match = re.search(r"(\d{3,6})\s*employees", content)
```

Activate
Go to Set

```
summary_sentences = content.split(". ")[:4] # Extract first 4 sentences

return InstitutionDetails(
    founder=founder_match.group(1) if founder_match else "Not Found",
    founded_year=founded_year_match.group(1) if founded_year_match else "Not Found",
    branches=branches_match if branches_match else ["Not Found"],
    employees=employees_match.group(1) if employees_match else "Not Found",
    summary=". ".join(summary_sentences)
)

# Step 3: Fetch details from Wikipedia
wiki = WikipediaQueryRun(api_wrapper=WikipediaAPIWrapper())
institution_name = "Apple Company"
wiki_content = wiki.run(institution_name)

# Step 4: Parse and display results
institution_details = parse_wikipedia_content(wiki_content)
print(institution_details.model_dump_json(indent=4))

{
    "founder": "Not Found",
    "founded_year": "Not Found",
    "branches": [
        "Not Found"
    ],
    "employees": "Not Found",
    "summary": "Page: Apple Inc.\nSummary: Apple Inc. is an American multinational corporation and technology company headquartered in Cupertino, California, in Silicon Valley. It is best known for its consumer electronics, software, and services. Founded in 1976 as Apple Computer Company by Steve Jobs, Steve Wozniak and Ronald Wayne, the company was incorporated by Jobs and Wozniak as Apple Computer, Inc"
}
```

Program 10:

Build a chatbot for the Indian Penal Code. We'll start by downloading the official Indian Penal Code document, and then we'll create a chatbot that can interact with it. Users will be able to ask questions about the Indian Penal Code and have a conversation with it.

```
In [1]: from langchain.text_splitter import RecursiveCharacterTextSplitter, SentenceTransfo
        import numpy as np # Import NumPy for numerical operations
        from pypdf import PdfReader # Import PdfReader to read PDF files
        from tqdm import tqdm # Import tqdm for progress bar visualization
```



```
In [2]: from pypdf import PdfReader # Import PdfReader from the pypdf library
        reader = PdfReader("BNS (IPC).pdf") # Create a PdfReader object to read the PDF fil
        pdf_texts = [p.extract_text().strip() for p in reader.pages] # Extract text from ea
        # Filter out empty strings from the list of extracted texts
        pdf_texts = [text for text in pdf_texts if text]
        print(pdf_texts[0]) # Print the word-wrapped version of the first text element extr
```

THE BHARA TIYA NY AYA SANHITA, 2023
NO. 45 OF2023

[25th December ,2023.]

An Act to consolidate and amend the provisions relating to offences and for matters connected therewith or incidental thereto.

Be it enacted by Parliament in the Seventy-fourth Year of the Republic of India as follows:—

CHAPTER I

PRELIMINARY

- 1.(1) This Act may be called the Bharatiya Nyaya Sanhita, 2023.
- (2) It shall come into force on such date as the Central Government may, by notification in the Official Gazette, appoint, and different dates may be appointed for different provisions of this Sanhita. Short title, commencement and

application.vlk/kkj.k

EXTRAORDINARY

Hkkx II —[k.M 1

PART II — Section 1

izkf/kdkj ls izdkf'kr

PUBLISHED BY AUTHORITY

lañ 53] ubZ fnYyh] lkseokj] fnlEcj 25] 2023@ ikS"k 4] 1945 %'kd%
No. 53] NEW DELHI, MONDAY, DECEMBER 25, 2023/PAUSH 4, 1945 (SAKA)

b1 Hkkx esa fHkUu i`"B la[;k nh tkrh gS ftlls fd ;g vyx ladyu ds :i esa j[kk tk ldsA Separate paging is given to this Part in order that it may be filed as a separate compilation. xxxGIDHxxx

xxxGIDExxx

jftLVa h lañ Mhñ ,yñ -(,u)04@0007@2003 —23 REGISTERED NO. DL-(N)04/0007/2003-23
MINISTRY OF LAW AND JUSTICE

(Legislative Department)

New Delhi, the 25th December, 2023/Pausa 4, 1945 (Saka)

The following Act of Parliament received the assent of the President on the 25th December, 2023 and is hereby published for general information:—

सी.जी.-डी.एल.-अ. -25122023-250883

CG-DL-E-25122023-250883

```
In [3]: from langchain.text_splitter import RecursiveCharacterTextSplitter, SentenceTransfo
character_splitter = RecursiveCharacterTextSplitter( # Initialize RecursiveCharacterTextSplitter
    separators=[ "\n\n", "\n", ". ", " ", ""], # Define separators to split text by
    chunk_size=1000, # Define the maximum chunk size in characters
    chunk_overlap=20 # Define the overlap between adjacent chunks in characters
)
character_split_texts = character_splitter.split_text('\n\n'.join(pdf_texts)) # Split the text into chunks
print(character_split_texts[10]) # Print the word-wrapped version of the 11th chunk
print(f"\nTotal chunks: {len(character_split_texts)}") # Print the total number of
```

Illustration.

A Magistrate exercising jurisdiction in respect of a charge on which he has power to sentence to fine or imprisonment, with or without appeal, is a Judge; (17) "life" means the life of a human being, unless the contrary appears from the context; (18) "local law" means a law applicable only to a particular part of India; (19) "man" means male human being of any age; (20) "month" and "year".--Wherever the word "month" or the word "year" is used, it is to be understood that the month or the year is to be reckoned according to the Gregorian calendar; (21) "movable property" includes property of every description, except land and things attached to the earth or permanently fastened to anything which is attached to the earth; (22) "number".--Unless the contrary appears from the context, words importing the singular number include the plural number, and words importing the plural number include the singular number;

Total chunks: 480

```
In [4]: from getpass import getpass # Import getpass for secure password input

#from Langchain import HuggingFaceHub # commented out line, likely an old import statement

from langchain_community.llms import HuggingFaceHub # Import HuggingFaceHub from Langchain
import os # Import os module for environment variables

inference_api_key = getpass() # Prompt user to enter their Hugging Face API key securely
#place your huggingface API key after running this cell
os.environ["HUGGINGFACEHUB_API_TOKEN"] = inference_api_key # Set the Hugging Face API token

inference_api_key # Display the API key (for debugging or confirmation, but generally not needed)

from langchain_community.embeddings import HuggingFaceInferenceAPIEmbeddings # Import HuggingFaceInferenceAPIEmbeddings
embedding_function = HuggingFaceInferenceAPIEmbeddings( # Initialize HuggingFaceInferenceAPIEmbeddings
    api_key=inference_api_key, # Pass the API key for authentication
    model_name="sentence-transformers/all-MiniLM-16-v2" #
)

from langchain_community.vectorstores import FAISS # Import FAISS for vector storage
db = FAISS.from_texts(character_split_texts, embedding_function) # Create a FAISS vector store

print(db.index.ntotal) # Print the total number of vectors in the FAISS index
```

.....

480

```
In [5]: query = "What does BNS Section 72 talks about ?" # Define the query string to search for

retrieved_documents = db.similarity_search(query) # Perform a similarity search in the database
# and store the retrieved documents
```

```
In [6]: retrieved_documents
```

```
Out[6]: [Document(id='98c47b0f-abc4-4f1a-a15e-1fccb0f3c139', metadata={}, page_content='victim.\n71.Whoever has been previously convicted of an offence punishable under section 64 or section 65 or section 66 or section 70 and is subsequently convicted of an offence punishable under any of the said sections shall be punished with imprisonment for\nlife which shall mean imprisonment for the remainder of that person's natural life, or with\ndeath.\n72.(1) Whoever prints or publishes the name or any matter which may make known\nthe identity of any person against whom an offence under section 64 or section 65 or\nsection 66 or section 67 or section 68 or section 69 or section 70 or section 71 is alleged or\nfound to have been committed (hereafter in this section referred to as the victim) shall be\npunished with imprisonment of either description for a term which may extend to two years\nand shall also be liable to fine.\n(2) Nothing in sub-section (1) extends to any printing or publication of the name or'),
```

```
Document(id='6f2c3cef-a48b-4646-abfd-b6879983b4bf', metadata={}, page_content='it is his duty\n to prevent.Sec. 1]THE GAZETTE OF INDIA EXTRAORDINARY21_____
```

```
'),
```

```
Document(id='e7b2b3f2-2d57-4c44-b13b-57d77e264ca0', metadata={}, page_content='Explanation.—For the purposes of this sub-section, “recognised welfare institution\nor organisation” means a social welfare institution or organisation recognised in this behalf\nby the Central Government or the State Government.\n73.Whoever prints or publishes any matter in relation to any proceeding before a\ncourt with respect to an offence referred to in section 72 without the previous permission of\nsuch court shall be punished with imprisonment of either description for a term which may\nextend to two years and shall also be liable to fine.\nExplanation.—The printing or publication of the judgment of any High Court or the\nSupreme Court does not amount to an offence within the meaning of this section.\nof criminal force and assault against woman\n74.Whoever assaults or uses criminal force to any woman, intending to outrage or\nknowing it to be likely that he will thereby outrage her modesty, shall be punished with'),
```

```
Document(id='c3f24224-9115-4009-9886-32b29d022bb0', metadata={}, page_content='THE BHARATIYA NYAYA SANHITA, 2023\nNO. 45 OF 2023\n[25th December , 2023.] An Act to consolidate and amend the provisions relating to offences and for\nmatters connected therewith or incidental thereto.\nBe it enacted by Parliament in the Seventy-fourth Year of the Republic of India as\nfollows:—\nCHAPTER I\nPRELIMINARY\n1.(1) This Act may be called the Bharatiya Nyaya Sanhita, 2023.\n(2) It shall come into force on such date as the Central Government may, by notification\nin the Official Gazette, appoint, and different dates maybe appointed for different provisions\nof this Sanhita. Short title,\ncommencement\nand\napplication.\n[Part II – Section 1\nnizkf/kdkj ls izdkf\\'kr\nPUBLISHED BY AUTHORITY\nnlañ 53] ubZ fnYyh] lkseokj] fnlEcj 25] 2023@ ikS\"k 4] 1945 %\\'kd%\nN EW DELHI, MONDAY, DECEMBER 25, 2023/PAUSH 4, 1945 (SAKA)\nbl Hkkx esa fHkUu i\"B la[;k nh tkrh gS ftlls fd ;g vyx ladyu ds :i esa j[kk tk ldsA'])]
```

```
In [ ]: import cohere
# Initialize the Cohere client
co = cohere.Client(api_key='eA50e27b8807ycXim4jLv5BxMJurbufz9e1AMJNz',) # Replace
```

```
In [22]: #In which section of BNS public servant is mentioned ?
#Which section of BNS explains religious riots?
#What is section 72 of BNS ?
#Which section talks about Waging war against the government? Explain with details
#What section of BNS talks about theft and robbery?
#which section of BNS explains about attempt to murder?
#Which section talks about Exploitation of a trafficked person?
```

Bharathiya Nyay Sanhita - IPC Chatbot

Without UI

```
In [14]: def rag(query, retrieved_documents, model="command-r-plus"):
    information = "\n\n".join([docs.page_content for docs in retrieved_documents])

    messages = [
        {
            "role": "system",
            "content": """You are a helpful expert in Bharatiya Nyay Sanhita (BNS). Your users are asking questions about information contained in Bharatiya Nyay Sanhita. You will be shown the user's question, and the relevant information from given documents. Note that if asked for section get BNS section number. Answer the user's question using only this information."""
        },
        {
            "role": "user",
            "content": f"Question: {query}. \n Information: {information}"
        }
    ]

    response = co.chat(
        model=model,
        message=query,
        documents=messages
    )

    return response.text

def chatbot(query):
    try:
        retrieved_documents = db.similarity_search(query, k=5)
        response = rag(query, retrieved_documents)
        source_text = "\n\n".join([doc.page_content for doc in retrieved_documents])
        return response, source_text
    except Exception as e:
        print("Error:", e)
        return str(e), ""

# Example interaction without UI
while True:
    query = input("Ask a question (or type 'exit'): ")
    if query.lower() == 'exit':
```

```
break
response, source_text = chatbot(query)
print("\n\n Response:", response)
print("\n\n Source Text:", source_text)
```

Ask a question (or type 'exit'): section 72

Response: Section 72 (1) states that whoever prints or publishes the name or any matter which may make known the identity of any person against whom an offence under section 64, 65, 66, 67, 68, 69, 70 or 71 is alleged or found to have been committed (referred to as the victim) shall be punished with imprisonment for up to two years and may also be fined.

Section 72 (2) states that nothing in sub-section (1) extends to any printing or publication of the name or matter in any recognised welfare institution or organisation.

Source Text: victim.

71.Whoever has been previously convicted of an offence punishable under section 64 or section 65 or section 66 or section 70 and is subsequently convicted of an

offence punishable under any of the said sections shall be punished with imprisonment for

life which shall mean imprisonment for the remainder of that person's natural life, or with death.

72.(1) Whoever prints or publishes the name or any matter which may make known the identity of any person against whom an offence under section 64 or section 65 or section 66 or section 67 or section 68 or section 69 or section 70 or section 71 is alleged or

found to have been committed (hereafter in this section referred to as the victim) shall be

punished with imprisonment of either description for a term which may extend to two years

and shall also be liable to fine.

(2) Nothing in sub-section (1) extends to any printing or publication of the name or

Explanation.—For the purposes of this sub-section, “recognised welfare institution or organisation” means a social welfare institution or organisation recognised in this behalf

by the Central Government or the State Government.

73.Whoever prints or publishes any matter in relation to any proceeding before a Court with respect to an offence referred to in section 72 without the previous permission of

such Court shall be punished with imprisonment of either description for a term which may

extend to two years and shall also be liable to fine.

Explanation.—The printing or publication of the judgment of any High Court or the Supreme Court does not amount to an offence within the meaning of this section.

Of criminal force and assault against woman

74.Whoever assaults or uses criminal force to any woman, intending to outrage or knowing it to be likely that he will thereby outrage her modesty, shall be punished with

(23) “oath” includes a solemn affirmation substituted by law for an oath, and any declaration required or authorised by law to be made before a public servant or to

be used for the purpose of proof, whether in a Court or not;

(24) “offence”.—Except in the Chapters and sections mentioned in sub-clauses (a) and (b), the word “offence” means a thing made punishable by this Sanhita, but--

(a) in Chapter III and in the following sections, namely, sub-sections (2), (3), (4) and (5) of section 8, sections 9, 49, 50, 52, 54, 55, 56, 57, 58, 59, 60, 61, 119,
 120, 123, sub-sections (7) and (8) of section 127, 222, 230, 231, 240, 248, 250, 251, 259, 260, 261, 262, 263, sub-sections (6) and (7) of section 308 and sub-section (2) of section 330, the word "offence" means a thing punishable under this Sanhita, or under any special law or local law; and
 (b) in sub-section (1) of section 189, sections 211, 212, 238, 239, 249, 253

by public

servant.62THE GAZETTE OF INDIA EXTRAORDINARY[Part II-_____

omission to apprehend on part of public servant bound to apprehend person under sentence or lawfully committed.

Sec. 1]THE GAZETTE OF INDIA EXTRAORDINARY71_____

Ask a question (or type 'exit'): exit

With Gradio UI

```
In [17]: #!pip install gradio
```

```
In [18]: import gradio as gr # Import Gradio for creating a user interface
```

```
In [19]: def rag(query, retrieved_documents, model="command"):
```

```
    """
```

```
    Performs Retrieval-Augmented Generation (RAG) to answer a query based on retrieved documents.
```

```
Args:
```

```
    query (str): The user's question or query.
```

```
    retrieved_documents (list): A list of documents retrieved from a vector database.
```

```
    model (str, optional): The Cohere chat model to use. Defaults to "command".
```

```
Returns:
```

```
    str: The generated answer from the Cohere chat model.
```

```
    """
```

```
# Extract the page content from each retrieved document and join them into a single string
information = "\n\n".join([docs.page_content for docs in retrieved_documents])
```

```
# Define the messages to be sent to the Cohere chat model.
```

```
# This includes a system message to set the context and a user message with the query.
```

```

messages = [
    {
        "role": "system",
        "content": """You are a helpful expert in Bharatiya Nyay Sanhita (BNS). Your users are asking questions about information contained in Bharatiya Nyay Sanhi You will be shown the user's question, and the relevant information from given docu Note that if asked for section get BNS section number. Answer the user's question using only this information.""" # System message to guid
    },
    {
        "role": "user",
        "content": f"Question: {query}. \n Information: {information}" # User message
    }
]

# Call the Cohere chat API to get a response based on the provided messages.
response = co.chat( # Using the 'co' Cohere client (assumed to be initialized earlier)
    model=model, # Specify the model to use
    message=query, # Pass the user's query as the main message
    documents=messages # Pass the list of messages including system and user previous messages
)

return response.text # Return the text of the generated response from the Cohere API

```

```

In [20]: # Define the chatbot function
def chatbot(query):
    """
    Chatbot function to answer user queries based on retrieved documents using RAG.

    Args:
        query (str): The user's question or query.

    Returns:
        tuple: A tuple containing the response from the RAG model and the source text.
               Returns an error message and empty source text if an exception occurs.
    """
    try:
        # Query FAISS to get retrieved documents
        retrieved_documents = db.similarity_search(query, k=5) # Perform similarity search

        # Debug: print retrieved documents
        # print("Retrieved Documents:", retrieved_documents) # Commented out debug

        # Call the RAG function
        response = rag(query, retrieved_documents) # Call the rag function (defined earlier)

        source_text = "\n\n".join([doc.page_content for doc in retrieved_documents])
        return response, source_text # Return the generated response and the combined source text

    except Exception as e: # Catch any exceptions that might occur during the process
        # Debug: print exception details
        print("Error:", e) # Print the error message to the console

    return str(e), "" # Return the error message as a string and an empty string

```

```
In [21]: # Set up the Gradio interface
iface = gr.Interface( # Create a Gradio Interface object

    fn=chatbot, # Pass the user query to the chatbot function. 'chatbot' is assumed
    inputs=gr.Textbox(lines=2, placeholder="Ask a Bharatiya Nyay Sanhita (BNS /IPC)

    outputs=[ # Define the output components as a list
        gr.Textbox(label="Response", lines=4), # First output is a textbox Labeled
        gr.Textbox(label="Source Text", lines=10) # Second output is a textbox Labe
    ],
    title="Bharatiya Nyay Sanhita", # Set the title of the Gradio interface.
    description="Ask any Bharathiya Nyay Sanhita related question, and I will provi
)

# Launch the Gradio interface
iface.launch() # Launch the Gradio interface to start the chatbot application.
```

Running on local URL: <http://127.0.0.1:7861>

To create a public link, set `share=True` in `launch()`.

