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

Composed by	Verified by	Approved by
	Mrs. MADHU NAGARAJ	Dr. ANITHA D B
	FACULTY COORDINATOR	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, 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.

9	Program 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	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.

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, 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.
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.

Lab Programs

Program 1:

Explore pre-trained word vectors. Explore word relationships using vector arithmetic. Perform arithmetic operations and analyze results.

```
!pip install gensim
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"))
```

Example 1: Find Similar Words

```
similar_to_mysore = model.similar_by_vector(model['bangalore'], topn=3)

print(f"Words similar to 'bangalore': {similar_to_mysore}")

result_vector_1 = model['waiter'] - model['man'] + model['woman']
result_1 = model.similar_by_vector(result_vector_1, topn=1)
print(f"'waiter - man + woman' = {result_1}")



# 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}")

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