Pranita Mongse

GENERATIVE AI ENGINEER

********16@gmail.com
pur, 44*, India

ABOUT ME

Driven Generative AI Engineer with a Master's in Computer Science, eager to contribute deep learning and NLP expertise to a pioneering team, while advancing in cutting-edge AI model development.

EDUCATION

BCA
G. H Raisoni Institute Of Information Technology
Nagpur
2010

MSc. Computer Science
G. H Raisoni Institute Of Information Technology
Nagpur
2014

SKILLS

Machine Learning and Deep Learning- Supervised and Unsupervised learning, Tensorflow, pytorch,keras

Natural Language Processing (NLP)-NLTK,Spacy,Hugging Face transformer

Programming Language-Python

Data Manipulation and Analysis-Pandas,Numpy

Neural Network-RNN,CNN

PROJECTS

Project Title: Generative Image Synthesis with CLIP and Taming Transformers

Developed a state-of-the-art image synthesis system that leverages advanced deep learning techniques to generate high-quality images from textual prompts. By integrating the CLIP (Contrastive Language-Image Pretraining) model for text and image understanding with Taming Transformers (specifically VQGAN), the project enables the creation of visually coherent and contextually relevant images based on user-defined descriptions. The system showcases the capabilities of generative AI to transform abstract concepts into concrete visuals, with applications in creative design, art, and content generation.

Tools & Libraries Used:

- Programming Languages: Python
- Deep Learning Libraries: PyTorch, PyTorch Lightning
- Computer Vision Libraries: OpenAI CLIP, Taming Transformers
- Data Handling: NumPy, Pandas
- Image Processing: PIL (Pillow), Matplotlib, ImageIO
- Dependency Management: OmegaConf, hfy, regex, tqdm, einops
- Environment: Google Colab (for GPU acceleration).

Key Responsibilities:

- Integrated Models: Combined OpenAI's CLIP and Taming Transformers to create a cohesive framework for text-to-image generation.
- Data Augmentation: Developed and implemented data augmentation techniques to enhance the diversity and quality of generated images.
- Parameter Optimization: Designed an optimization strategy using the AdamW algorithm to fine-tune latent representations, improving image synthesis quality.
- Image Generation Pipeline: Built a pipeline for generating images from encoded text prompts, ensuring high-resolution output and coherence with input descriptions.
- Crops Generation: Created an innovative approach for generating augmented image crops to enrich the training dataset and improve model performance.
- Training Execution: Managed training sessions, iterating on hyperparameters, monitoring loss metrics, and adjusting strategies to maximize image fidelity.
- Visualization Tools: Implemented tools for visualizing generated images and the optimization process, enabling better analysis and iterative refinement.
- Interpolation Feature: Developed a feature to create smooth transitions between generated images, enhancing the visual storytelling aspect of the project.

Computer Vision and image processing-opencv,PII,YOLO8

Generative Models - GANs, VAE, Autoregressive model,BERT, LLAMA, OLLAMA,GROQ

Data Visualization - Matplotlib,seaborn,Tableau,PowerBI

LINKS

LinkedIn:

www.linkedin.com/in/pr******gse16

LANGUAGES

English

Marathi

Hindi

Project Title: Integration of GROQ, GEMMA, and LangChain for Intelligent Document Processing

Developed a robust document processing pipeline leveraging the capabilities of GROQ, GEMMA, and LangChain to extract insights from a text document. The project included embedding generation, similarity searching, and question answering based on the processed document.

Tools and Libraries Used:

- Programming Languages: Python
- Libraries: LangChain, Hugging Face Transformers,FAISS, Chromadb, Unstructured, Groq, ImageIO
- Development Environment: Google Colab

Key Responsibilities:

- Environment Setup:Configured the development environment by installing necessary libraries,including LangChain and its dependencies.
- Document Loading:Implemented a document loader to read and load content from a specified text file, ensuring the information is accessible for processing.
- Text Wrapping Utility:Created a function to format and preserve the readability of text, maintaining original newlines while wrapping lines to a specified width.
- Text Splitting Utilized LangChain's CharacterTextSplitter to partition the loaded document into manageable chunks, facilitating efficient processing and retrieval.
- Embedding Generation:Integrated Hugging Face embeddings to convert text chunks into vector representations for similarity searches.
- Vector Store Implementation:Established a FAISS vector store to manage and perform similarity searches on the embedded documents, optimizing retrieval of relevant information.
- Query Execution: Executed queries against the vector store to retrieve contextually relevant document sections for further analysis.
- Question Answering System: Implemented a question-answering chain using the ChatGroq model to extract specific information from the documents based on user queries.
- Performance Optimization:Optimized the performance of the document processing pipeline by refining the embedding and similarity search mechanisms.

Project Title:Web Scraping Job Listings

Developed a comprehensive web scraping application to extract and analyze job listings from TimesJobs, focusing on data science positions.

- Technologies Used: Python, BeautifulSoup, Pandas, Requests, Matplotlib, Seaborn, WordCloud

Key Responsibilities:

- Implemented functions to extract job titles, companies, skills, locations, and salary information, resulting in a dataset of 250 unique job listings.
- Conducted data cleaning, including handling duplicates and missing values, ensuring the dataset was accurate and reliable for analysis.
- Visualized key insights through various plots, including:
- Word cloud of in-demand skills.
- Bar charts showing job openings by city and top companies offering internships.

- Salary distribution histogram and experience level distribution.
- Analyzed the correlation between salary and experience required, providing insights into market trends.

Project Title: Interactive Chatbot using Llama3 and Streamlit

Developed an interactive chatbot application utilizing the Llama3 language model to facilitate real-time natural language processing. The application allows users to input text prompts and receive instant, contextually relevant responses, simulating a conversational experience. The design emphasizes user engagement through a clean interface and responsive feedback mechanisms.

Technologies Used:

- Programming Language: Python
- Frameworks/Libraries:
- langchain_community for Llama3 model integration
- Streamlit for building the web application interface
- Model: Llama3 (a state-of-the-art language model for generating human-like text).

Key Responsibilities:

- User Interface Development: Designed a user-friendly interface using Streamlit, featuring an input area for users to enter prompts and a button to generate responses.
- Natural Language Processing Integration:Utilized the Llama3 model from the langchain_community library to process user inputs and generate contextually relevant responses.
- Real-time Response Generation: Implemented a dynamic response generation feature, where the application streams responses back to the user progressively, enhancing the interactive experience.
- Progress Indicator: Incorporated a spinner to indicate processing time while the model generates responses, improving user engagement and feedback.
- Error Handling and Robustness: Developed robust error handling to manage potential issues during response generation, ensuring the application remains stable and user-friendly.