Vidya Search Engine

Author : Ranamalla Nithin Reddy **Date :** January 5, 2025

Project Overview

The Vidya Search Engine is a specialized search engine developed to extract and rank relevant content from various online sources, focusing on data science and machine learning topics. It integrates natural language processing (NLP) and machine learning models to deliver accurate and insightful results. This project is deployed on Hugging Face Spaces.

Key Features

- **Search Relevance**: Uses semantic search to rank results.

  - **Deployment**: Hosted on Hugging Face Spaces for easy accessibility.

  - **Pre-trained Models**: Employs BERT and Sentence-Transformers for contextual understanding.

  - **Web Scraping**: Utilizes BeautifulSoup to extract relevant articles and data from websites.

  - **Interactive Interface**: Powered by Streamlit for a user-friendly UI.

Approach and Steps

Step 1: Problem Definition
- Understand the need for a search engine tailored to data science and machine learning topics.
- Define the objectives: relevance, speed, and user accessibility.

Step 2: Environment Setup
1. Create a Python virtual environment:
`bash<br> python -m venv venv<br>`
2. Activate the virtual environment:
- Windows: `venv\Scripts\activate`
- Mac/Linux: `source venv/bin/activate`
3. Install required libraries:
`bash<br> pip install -r requirements.txt<br>`

Step 3: Data Collection
- Use web scraping techniques with BeautifulSoup to gather data from selected sources.
- Save data in a structured format (e.g., JSON or CSV).

Step 4: Model Selection and Implementation
- Choose a pre-trained Sentence-Transformer model for semantic search.
- Integrate FAISS for efficient similarity search.
- Use Scikit-learn to preprocess and vectorize text data.

Step 5: Backend Development
- Create a Python backend to:
- Load and preprocess data.
- Compute embeddings using the Sentence-Transformers model.
- Rank results using cosine similarity.

Step 6: UI Development
- Develop the frontend using Streamlit:
- Input field for user queries.
- Display ranked results interactively.

Step 7: Deployment
- Deploy the application on Hugging Face Spaces.
- Ensure all dependencies are listed in `requirements.txt`.

Step 8: Testing and Feedback
- Test the application for performance and usability.
- Collect user feedback to refine features.

Requirements

Libraries and Tools
- **Streamlit: For UI development.**
- **Sentence-Transformers: For embedding generation.**
- **FAISS: For similarity search.**
- **BeautifulSoup: For web scraping.**
- **Scikit-learn: For data preprocessing.**

Installation
Run the following command to install all dependencies:
`bash<br>pip install -r requirements.txt<br>`

Example `requirements.txt`
`<br>streamlit==1.41.1<br>sentence-transformers==3.3.1<br>faiss-cpu==1.7.4<br>beautifulsoup4==4.12.3<br>requests==2.32.3<br>numpy==1.2learn==1.6.0<br>scipy==1.10.1<br>`

Deployment

- Clone the repository:
  `bash<br> git clone <repository_url><br>`
  - Navigate to the project directory:
  `bash<br> cd vidya-search-engine<br>`
  - Run the application locally:
  `bash<br> streamlit run app.py<br>`

How to Access the Deployed Application

Online Access
- **Hugging Face Spaces: Vidya Search Engine**

Source Code
- **GitHub Repository: Vidya Search Engine**

Contact

For any issues or suggestions, please contact:
- **Name**: Ranamalla Nithin Reddy
- **Email**: reddynithin957@gmail.com
- **Phone**: 8179869350