TRU Recommend: Project Recommendation System

Project Overview

TRU Recommend is a recommendation system designed to match students with suitable projects and recommend students to clients based on project requirements. Leveraging advanced Natural Language Processing (NLP) models, the system generates embeddings from textual descriptions of skills, tools, certifications, past projects, and project requirements. The main goal is to streamline the process of matching students to projects, ensuring the best fit based on their profiles.

Objectives

- **Personalized Project Recommendations:** Help students find projects that align with their skills and experiences.
- **Client Recommendations:** Assist clients in identifying suitable candidates for their projects based on project descriptions.

Methodology

1. **Data Collection:**
    - **Resumes Dataset:** Contains information about students' skills, tools, certifications, and past projects.
    - **Projects Dataset:** Contains information about project descriptions and required skills.
    - **Resume Samples:** Contains 10 resume samples as pdf that are used for testing and demonstrating the functionality of the TRU Recommend system
2. **Data Processing:**
    - Combined relevant features from both datasets to create a unified text representation.
    - Used pre-trained NLP models (Bert, GPT2, Roberta) to generate embeddings for these text representations.
3. **Embedding Generation:**
    - Loaded pre-trained models and tokenizers.
    - Generated embeddings for combined text fields of both resumes and projects.
4. **Recommendation Algorithm:**
    - Utilized cosine similarity to find the best matches between student profiles and project requirements.
    - Implemented functions to recommend projects to students and students to clients.

Models Used

- **Bert:** bert-base-uncased
- **GPT2:** gpt2
- **Roberta:** roberta-base

File Structure and Description

```
TRU_Recommend/
├── app.py
├── utility.py
├── resumeextract.py
├── generate_embedding.py
├── datasets/
│   ├── resumes.csv
│   ├── projects.csv
│   ├── resume_samples/  # Contains 10 sample resume PDFs
└── embedding_pickelfile/
│      ├── resumes_with_embeddings.pkl
│      └── projects_with_embeddings.pkl
```

1. app.py

The main file that runs the Streamlit web application. This file handles the user interface and interactions. Depending on the selected mode (User Mode or Client Mode), it provides different functionalities for recommending projects to students or students to clients.

- **User Mode:**
  - Allows students to upload their resumes or manually input their skills, tools, certifications, and past projects.
  - Provides recommendations for projects based on the user's profile.
- **Client Mode:**
  - Allows clients to select a project.
  - Provides recommendations for students suitable for the selected project.

2. utility.py

Contains utility functions and pre-trained model loading necessary for the recommendation system.

- **Functions:**

    - get_embeddings: Generates embeddings for the given text using the specified tokenizer and model.
    - recommend_students: Recommends students for a given project based on embeddings and cosine similarity.
    - recommend_projects: Recommends projects for a given student based on embeddings and cosine similarity.
- **Models and Tokenizers:**
    - Loads and initializes the Bert, GPT2, and Roberta models and their respective tokenizers.
- **Skills, Tools, Certifications, and Projects Lists:**
    - Lists of common skills, tools, certifications, and projects used in the multiselect widgets in the Streamlit app.

3. get_embedding.py

Script to preprocess the datasets and generate embeddings for resumes and projects using the pre-trained models.

- **Steps:**
    - Loads the resumes and projects datasets.
    - Combines relevant text features to create a unified text representation for both resumes and projects.
    - Generates embeddings for the combined text fields using the pre-trained models.
    - Saves the generated embeddings to pickle files for later use.

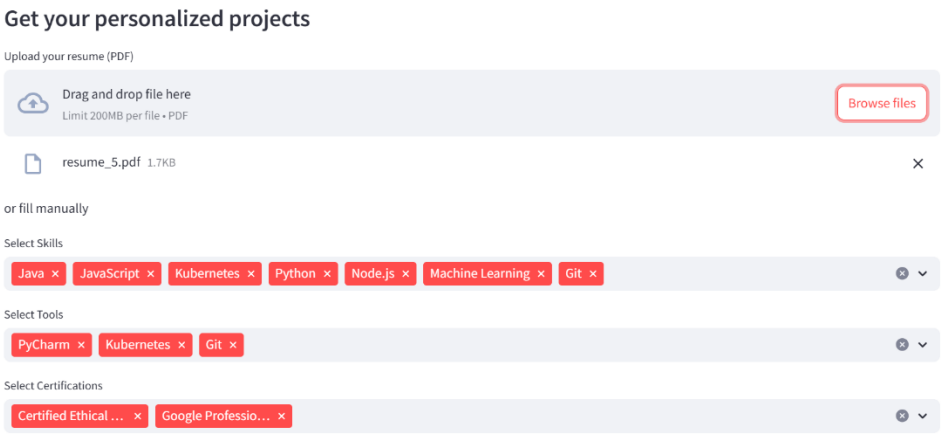
4.resumeextract.py

    - Reads the text from each page of the PDF resumes.
    - Finds and extracts mentions of skills, tools, and certifications from the resume text.

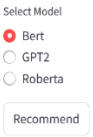
Web Application/ Streamlit UI

Dashboard

User Mode



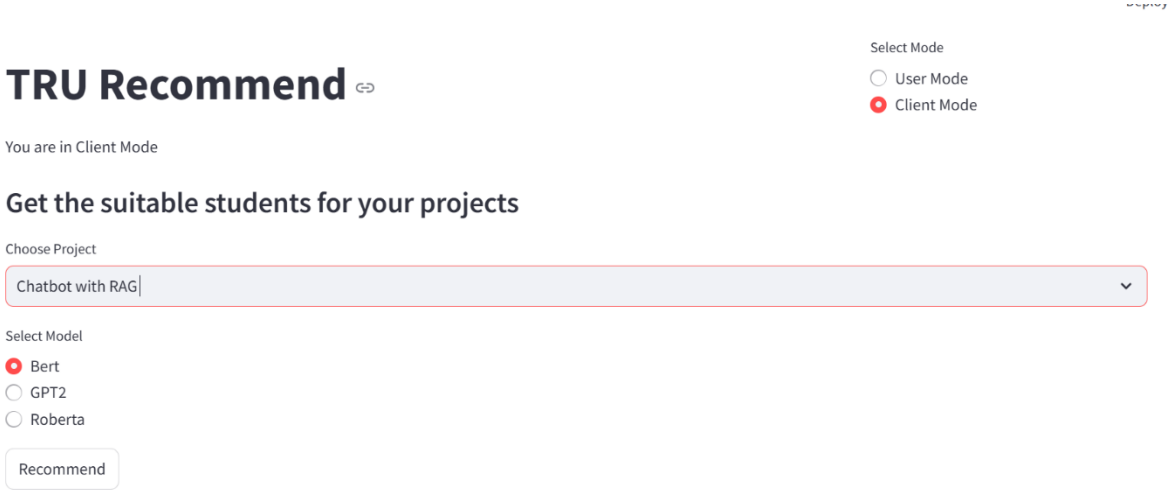
Model Selection



Results

	Project_ID	Project_Name	Project_Description	Skills_Required
0	28	AI-Powered Chat Application	Develop a chat application with AI-powered features like smart replies.	Python, NLP, Web Development, WebSockets
1	34	Real-Time Chat Application	Build a real-time chat application with user authentication.	HTML, CSS, JavaScript, Node.js, WebSockets
2	18	E-learning Platform	Create a platform for online courses with video lectures and quizzes.	HTML, CSS, JavaScript, Node.js, MongoDB
3	6	AI-Generated Art	Create an application that generates artwork using Generative Adversarial Networks (GANs).	Python, Deep Learning, GANs, Computer Vision
4	40	AR-Based Interior Design App	Create an augmented reality app to visualize interior design options.	Java/Kotlin (Android), Swift (iOS), ARKit/ARCore, Mobile Development
5	31	Cloud-Based File Storage System	Build a secure cloud-based file storage and sharing system.	Python, Cloud Computing, AWS/GCP/Azure, Web Development
6	13	Personal Finance Manager	Develop a web application to manage personal finances and track expenses.	HTML, CSS, JavaScript, React, Node.js
7	12	Cloud Cost Optimization Tool	Create a tool to optimize cloud resource usage and reduce costs.	Python, Cloud Computing, AWS/GCP/Azure, Data Analysis
8	41	Online Quiz Platform	Build a platform for creating and taking online quizzes.	HTML, CSS, JavaScript, React, Node.js

Client Mode



Results

Recommended students for your projects:

	Student_ID	Skills	Tools	Certification	Past_Project_Descriptions
0	Daniel Morgan	Python, Data Analysis, Machine Learning	Jupyter Notebook, Pandas, Scikit-Learn	Data Science Professional Certificate, Machine Learning by Stanford University	Sentiment Analysis Tool, Automated Trading Bot
1	Wendy Campbell	Python, Data Analysis, Machine Learning	Jupyter Notebook, Pandas, Scikit-Learn	Data Science Professional Certificate, Machine Learning by Stanford University	Sentiment Analysis Tool, Automated Trading Bot
2	Isabella Harris	Python, Data Analysis, Machine Learning	Jupyter Notebook, Pandas, Scikit-Learn	Data Science Professional Certificate, Machine Learning by Stanford University	Sentiment Analysis Tool, Automated Trading Bot
3	Paul Lopez	Python, Data Analysis, Machine Learning	Jupyter Notebook, Pandas, Scikit-Learn	Data Science Professional Certificate, Machine Learning by Stanford University	Sentiment Analysis Tool, Automated Trading Bot
4	Alice Johnson	Python, Data Analysis, Machine Learning	Jupyter Notebook, Pandas, Scikit-Learn	Data Science Professional Certificate, Machine Learning by Stanford University	Sentiment Analysis Tool, Automated Trading Bot
5	Jack Robinson	JavaScript, Angular, Node.js	Visual Studio Code, Git, npm	Full-Stack Web Development by Coursera, Angular Developer Certification	Personal Portfolio Website, Real-Time Chat Application
6	Quinn Hill	JavaScript, Angular, Node.js	Visual Studio Code, Git, npm	Full-Stack Web Development by Coursera, Angular Developer Certification	Personal Portfolio Website, Real-Time Chat Application
7	Xavier Parker	JavaScript, Angular, Node.js	Visual Studio Code, Git, npm	Full-Stack Web Development by Coursera, Angular Developer Certification	Personal Portfolio Website, Real-Time Chat Application

Conclusion

TRU Recommend demonstrates the potential of using advanced NLP techniques to create a recommendation system that benefits both students and clients. By leveraging pre-trained models and embedding techniques, the system provides accurate and meaningful recommendations, helping users make informed decisions about project assignments and candidate selection.