Project Report: Conference Classifier

Overview

The "Conference Classifier" is a machine learning-driven system designed to classify research papers into one of five prestigious conferences: **CVPR, NeurIPS, EMNLP, TMLR, and KDD**. The tool employs Google Generative AI (Gemini model) to evaluate the suitability of a given document based on its content, methodology, and findings. The application provides not only a classification but also a rationale for why the paper is a good fit for the identified conference.

Features

- **PDF and Text Input**: Accepts research papers either as uploaded PDF files or pasted text content.
- **Automated Classification**: Utilizes Google Generative AI to classify the document into one of the five conferences.
- **Rationale Generation**: Provides a concise explanation for the classification decision.
- **User-Friendly Interface**: Developed using Streamlit for an intuitive and interactive user experience.

Project Structure

The project is organized into a modular structure with a clear separation of concerns, as follows:

project-root

├— src

| ├— __init__.py

| ├— Authenticate_docs.py

| ├— Pdf_fetcher.py

| ├— process.py

├— .env

├— .gitignore

├— LICENSE

├— main.py

├— README.md

├— requirements.txt

├— results.csv

src Folder

The src folder is the core of the project, containing all the logic for paper classification:

1. **Authenticate_docs.py**:

    - Handles authentication and API interactions.
    - Manages secure access to external APIs, such as Google Generative AI, using environment variables.
2. **Pdf_fetcher.py**:
    - Contains utility functions to process uploaded PDF files.
    - Extracts text from PDF documents using libraries like PyPDF2 or PyMuPDF.
3. **process.py**:
    - Implements preprocessing functions to sanitize and prepare document text.
    - Handles special characters, formatting, and ensures compatibility with AI models.
4. **__init__.py**:
    - Initializes the src module, enabling imports across the project.

Other Files

- **.env**: Stores environment variables, such as the API key for Google Generative AI.
- **.gitignore**: Excludes sensitive files like .env and temporary files from version control.
- **main.py**: Entry point for the Streamlit application.
- **README.md**: Comprehensive documentation for the project.
- **requirements.txt**: Lists all dependencies required for the project.
- **results.csv**: Stores classification results for analysis or future reference.

Workflow

1. **User Input**:
    - Upload a PDF file or paste the text content of a research paper.
2. **Preprocessing**:
    - If a PDF is uploaded, Pdf_fetcher.py extracts text from the document.
    - The extracted or pasted text is sanitized by process.py.
3. **Classification**:
    - The sanitized text is passed to the classify_conference function in Authenticate_docs.py.
    - Google Generative AI evaluates the document and returns a classification with rationale.
4. **Output**:
    - The application displays the classified conference and its rationale to the user.

Technologies Used

- **Programming Language**: Python
- **Frontend Framework**: Streamlit
- **AI Model**: Google Generative AI (Gemini model)
- **Libraries**: PyPDF2, os, dotenv, LangChain, etc.

Results

The tool successfully classifies research papers into appropriate conferences and provides meaningful rationales. It streamlines the process of determining the best venue for academic publications, saving researchers time and effort.

Future Improvements

- Incorporate additional conferences to widen the scope.
- Enhance PDF parsing to support more complex document structures.
- Add multilingual support for non-English research papers.

Conclusion

The Conference Classifier is a robust application designed to assist researchers in identifying the most suitable conferences for their work. By leveraging AI, it ensures accuracy and efficiency in the classification process.