Enhancing Undergraduate Research Recruitment through NLP-Driven Application Matching

Carl Bennett III
Department of Computer Science
Tuskegee University
Tuskegee, Alabama
cbennett5173@tuskegee edu

ABSTRACT

In the current academic landscape, the bridge between undergraduate talent and faculty research projects is often built on informal emails or static forms. For professors, reviewing dozens of statements of interest to find specific technical overlaps is time-consuming and ineffective. For students, the lack of transparency in opportunities and how their skills align with them can be a barrier to entry. This project addresses these inefficiencies by providing a centralized, intelligent platform that digitizes recruitment and applies computational linguistics to assist in decision-making.

This project follows a decoupled, three-tier architecture designed for scalability and separation of concerns. The frontend utilizes React.js for a dynamic User Interface (UI) that provides role-based dashboards. It handles state management for real-time application tracking and provides a responsive environment for both students and professors/Principal Investigators. The core of the project lies in the backend API, constructed with the Django REST Framework. Django manages the authentication logic, serves as the API gateway, and hosts the Machine Learning (ML) utility scripts. The relational database, MySQL, was chosen for its ACID compliance. This ensures that sensitive student data and application records remain consistent and secure. The core feature of this portal is the integration of the scikit-learn library to perform semantic analysis on text data. The matching process follows a three-stage pipeline: Pre-processing, Vectorization, and Similarity Scoring. To compare a professor's project description with a student's statement and Resume, the text must be converted into numerical vectors. This is accomplished by using **Term Frequency-Inverse Document Frequency (TF-IDF) (1).**

$$W_{t,d} = tf_{t,d} \cdot log(\frac{N}{df_t}) \quad (1)$$

Where $tf_{t,d}$ is the frequency of term t in document d, and the second term is the Inverse Document Frequency, which penalizes common words (like "the" or "and") while rewarding specific technical terms (like "Python" or "Microbiology").

Once the text is vectorized, the **Cosine Similarity (2)** is calculated to determine the distance between the two vectors. This measures the cosine of the angle between them in a high dimensional space:

$$\text{similarity} = \cos(\theta) = \frac{\text{A} \cdot \text{B}}{||\text{A}||\,||\text{B}||} \quad (2)$$

A score of 1.0 (100%) indicates a perfect keyword alignment, while 0.0 indicates no overlap.

This project implements several key features to enhance the user experience. By using Role-Based Access Control, Users are automatically directed to either the Student or Professor dashboard upon login. Keywords in a student's statement of interest or resume are highlighted using the TF-IDF feature names. The system identifies the top 5 overlapping terms between the student and the project, providing a quick, at-a-glance description of student suitability. Applications can be sorted by their keyword alignment immediately, allowing professors to prioritize the most relevant candidates.

By moving beyond manual review and adopting NLP-driven matching, this project reduces the administrative friction in undergraduate research. Future iterations of this work aim to incorporate Large Language Models (LLMs) such as Gemini or GPT-4 to provide nuanced summaries of student applications and support multilingual sentiment analysis for interdisciplinary projects.

KEYWORDS

Natural Language Processing, TF-IDF Vectorization, Cosine Similarity, Full-Stack Web Development, Machine Learning