Abstract

This project proposes an approach to help students in computer school of the University of Leeds in finding suitable supervisors for their final projects. Leveraging multiple machine learning and NLP techniques, this project offers a system that recommends supervisors based on user input, generates summaries for each potential supervisor and recommend similar supervisors to users. The methodologies encompassed include various vectorization techniques, recommendation algorithms, text generation models, topic modeling, and clustering recommendation. Using the OpenAI API for vector generation and text generation further enhanced the system's performance. The study illustrates the strengths and weaknesses of each approach and indicates the most effective techniques for the specific context of the project.

Introduction

The process of finding an appropriate supervisor of their final projects can be confusing and time-consuming for students. Currently students have to read through tutor profiles one by one, and given that the School of Computing now has 101 tutors, this process takes at least an hour, and if the number of supervisors is increased in the future, they will spend more time on selecting supervisors. And the process of selecting supervisors was complex. After reading through a large number of profiles, students also need to compare and rank supervisors, then they can submit their choice.

This necessitates an effective system that can alleviate this burden and optimize the supervisor selection process. This project aims to address this issue by developing a recommendation system which can reduce the time of choosing supervisors, help students ranking the supervisors and identify similar supervisors for students.

This project involves an array of methodologies and technologies. The project starts with data collection, where web scraping technique is employed to gather data from the School of Computing's page. Data cleaning methods are then utilized to ensure the data quality and prepared for testing different vectorization methods.

Different vectorization methods, such as Word2Vec, Doc2Vec, CountVectorizer, and TF-IDF, are tested and evaluated. These vectorized data are then employed in various recommendation approaches, cosine similarity is one of the most prominent. Another approach, Jaccard similarity also tested. Furthermore, text generation models such as T5, Bart, and TextRank are explored to generate brief introductions of all supervisors.

Topic modeling techniques, namely Latent Dirichlet Allocation (LDA), Non-negative Matrix Factorization (NMF), and BERTopic, are also examined in this project to recommend supervisors by topics. Exploration of these techniques gives the idea of using OpenAI's model for assistance, followed by testing and evaluation of OpenAI's text (topic) generation and vectorization models. Lastly, clustering recommendation (KMeans clustering) are evaluated and used to recommend supervisors with similar backgrounds.

Throughout this project, continually evaluate the performance of different method to identify the most effective techniques for recommendation system. By elucidating these processes, this project aims to provide methodological ideas and technical supports for the future establishment of a supervisor recommendation system.

Aim

The principal aim of this project is to develop a recommendation system to aid the computer science students at the University of Leeds in identifying suitable supervisors for their final year projects. The project also seeks to analyze the performance of various recommendation strategies, vectorization methods, and text generation models, with the goal of providing a comprehensive, user-friendly tool that recommends suitable supervisors based on user input.

Objectives

1. To obtain and clean data about faculty members in the School of Computer Science from the University of Leeds' website.

2. To evaluate various vectorization techniques such as Word2Vec, Doc2Vec, CountVectorizer, and TF-IDF for their efficiency and applicability to our dataset.

3. To implement and compare different recommendation methods including cosine similarity and Jaccard similarity to identify the most effective one.

4. To generate brief introductions for each faculty member using text generation models like T5, Bart, or TextRank.

5. To explore topic modeling techniques (LDA, NMF, BERTopic) and OpenAI APIs (text-davinci-003 and text-embedding-ada-002) for enhancing the recommendation system.

6. To design a cluster-based recommendation approach that recommends supervisors with similar research topics or backgrounds.

Deliverables

1. A cleaned and processed dataset of faculty members in the School of Computer Science at the University of Leeds, including their profiles, areas of research interest, and published works.

2. A comparative study of different vectorization techniques and recommendation methods, along with their strengths, weaknesses, and the rationale for the selected method.

3. A functional recommendation system that takes user input and outputs a list of recommended supervisors.

4. Generated text summaries for each faculty member, providing a brief introduction based on the data obtained.

5. A comprehensive evaluation of different topic modeling techniques and OpenAI APIs, with justifications for the chosen ones.

6. Implementation of a clustering recommendation system, providing an alternative way for users to find suitable supervisors.

7. A final report documenting the entire process, findings, and future directions for improvement.

Methodology

This project employs a hybrid research methodology encompassing elements from data science, machine learning, and natural language processing. The key stages of this process include data scraping and cleaning, feature vectorization, similarity computation, text generation, topic modeling, and cluster-based recommendation.

1. Data Scraping and Cleaning: This process begins with the extraction of information from the web pages of staff members in the School of Computer Science at the University of Leeds, employing web scraping techniques. The data harvested include individual profiles, areas of research interest, and published works. Following this, we undertake a comprehensive data cleaning process, which involves the removal of HTML symbols, abbreviation expansion, exclusion of generic terminologies, and handling of empty profiles. The ultimate objective of this phase is to ensure that the data, which form the input for subsequent stages, are clean, accurate, and comprehensive.

2. Feature Vectorization: Once the data are suitably cleaned, different vectorization techniques, including Word2Vec, Doc2Vec, CountVectorizer, and TF-IDF, are explored. These techniques are used to transform the textual data into vector format, suitable for machine learning algorithms. Each method is evaluated on its performance, with the most effective method selected for incorporation into our project.

3. Similarity Computation: Built on the vectorized data, we implement a cosine similarity computational approach to compare the similarities between different supervisors. This phase aids the system in recommending supervisors most relevant to user queries. Furthermore, other recommendation approaches such as the Jaccard similarity are tested in order to identify the optimal recommendation strategy.

4. Text Generation: To provide a comprehensive understanding of each supervisor, text generation models such as T5, Bart, and TextRank are employed to create a brief introduction for each supervisor. These models are compared on their performance with the highest-quality model being selected for our project.

5. Topic Modeling: In this stage, we leverage topic modeling techniques, such as Latent Dirichlet Allocation (LDA), Non-negative Matrix Factorization (NMF), and BERTopic to investigate the research fields of the supervisors. The goal is to recommend supervisors based on the similarity of their research topics. Additionally, APIs from OpenAI, including text-davinci-003 and text-embedding-ada-002, are utilized to further refine our recommendation results.

6. Clustering Recommendation: The final stage of our project involves experimentation with a cluster-based recommendation approach. We perform KMeans clustering on the research topics and backgrounds of supervisors, aiming to recommend supervisors with similar research fields or backgrounds. This form of recommendation is deployed when a user has chosen a supervisor, presenting other supervisors in the same cluster as additional options.