GALALA UNIVERSITY
Powered by Arizona State University®

Book Blurb Recommender

Name	ID
Samaa Khaled Mohamed	222100761
Habiba Ahmed Abdelnaby	222100471
Youssef Husseiny Fathy	222101943
Mohamed Desoky Ibrahim	222101362
Nada Gamal ElAdl	222101750

Project Overview

This project enhances book recommendation systems using NLP to analyze a book's blurb—a concise summary highlighting key characters and conflicts. Unlike traditional methods relying on lengthy descriptions, our approach focuses on the blurb for faster processing and improved efficiency without sacrificing relevance. This streamlined method offers users quick and accurate book suggestions, making the discovery process more intuitive and enjoyable

---

Introduction

In the age of information overload, finding the perfect book to read can be a daunting task. To address this challenge, we propose an intelligent book recommender system powered by Natural Language Processing (NLP). This project aims to revolutionize the recommendation process by focusing on an underutilized but impactful source of information—the book's blurb. Unlike traditional methods that rely on lengthy and detailed descriptions, our approach extracts key features from the concise yet insightful blurb, which often highlights the main characters and driving conflicts of the story. By processing this succinct text, we aim to enhance computational efficiency while maintaining, or even improving, the relevancy of recommendations. This streamlined approach not only reduces the processing load but also aligns with the end-user's natural reading patterns, ensuring a seamless and enjoyable discovery experience

---

Project Features

1.Interface Design:

- This project includes an intuitive user interface developed using HTML and CSS The interface allows users to interact seamlessly with the backend, facilitating data input, parameter selection, and results visualization
- The interface is designed to be user-friendly, offering real-time feedback and error handling for a smooth user experience

2.Backend Implementation:

The backend is implemented using Django, leveraging its powerful web framework capabilities for handling computations, data transformations, and result generation. Django's scalability ensures that the pipeline can handle large datasets efficiently

---

Methodology

1. Data Preprocessing:

- **Column Selection:**
  Non-essential columns such as book_format, book_authors, and book_isbn were removed.

- **Missing Data Handling:**

  Missing descriptions were filled using the corresponding book titles.

- **Deduplication:**
  Duplicate book titles were removed, and descriptions were aggregated using the group by function.

- **Text Cleaning:**
  Converted text to lowercase
  Removed non-alphanumeric characters
  Tokenized text using NLTK

- **Stopword Removal:**
  Stop words from the NLTK library were removed for texts longer than five words.

- **Lemmatization:**
  Words were lemmatized to their base forms using WordNetLemmatizer.

- **Language Translation:**
  Non-English descriptions were translated into English using Google Translate.

  Language detection ensured consistency and avoided translation of numeric or non-linguistic texts.

- **Text Improvement:**
  Short or incomplete descriptions were augmented using the book title or cleaned of invalid patterns such as excessive punctuation.

2. Feature Extraction:

TF-IDF Vectorization:

- Applied TF-IDF vectorization on cleaned descriptions.
- Configured with parameters:
- Maximum features: 10,000

- Minimum document frequency: 2
- Maximum document frequency: 0.9
- N-grams: (1, 3)
- Sublinear term frequency scaling.

Sentence Embedding:

Used the Sentence Transformer model all-mpnet-base-v2 to encode book descriptions into high-dimensional vectors. This allowed for semantic similarity comparisons using cosine similarity.

3. Recommendation Algorithms:

cosine-Based Recommendation

- **Input Preprocessing:**
  User-provided blurbs were cleaned and vectorized using the same preprocessing pipeline as the dataset.
- **Similarity Calculation:**
  Calculated cosine similarity between the user vector and all book vectors.
- **Top-N Selection:**
  Retrieved the top N books with the highest similarity scores.

Transformer-Based

- **Recommendation User Blurb Encoding:**
  The Sentence Transformer model encoded the user blurb into an embedding vector.
- **Cosine Similarity:**
  Calculated pairwise cosine similarity between the user embedding and book embeddings.
- **Ranking and Selection:**
  Ranked books based on similarity scores and returned the top N results

---

Results

The preprocessing pipeline demonstrated significant improvements in data quality and readiness for analysis. Key results include:

- Reduced processing time for large datasets by automating repetitive tasks.
- Enhanced data integrity through consistent cleaning methods and error handling.
- Improved model performance by delivering well-preprocessed datasets, as evidenced by better metrics in machine learning experiments.

Technologies Used

- **Frontend:** HTML and CSS for building the user interface, ensuring accessibility and responsiveness.
- **Backend:**
  - Django for implementing the preprocessing logic and algorithms, enabling seamless integration and scalability.
  - Libraries: NumPy, pandas, scikit-learn, matplotlib for efficient data handling and visualization.

Application:

This project serves as the foundation for an intelligent book recommender system with a user-friendly interface built using HTML and CSS. The interface allows users to input a book title or select preferences, presenting recommendations based on blurb analysis.The backend, developed with Django, handles data processing and manages the recommendation algorithm. It extracts features from book blurbs, compares them, and generates personalized suggestions efficiently. This combination of a clean front-end design and a robust back-end ensures a seamless and engaging user experience

---

Conclusion

By combining an HTML/CSS interface with a Django backend and leveraging NLP for text analysis, this project provides a user-friendly and efficient book recommendation system. NLP plays a crucial role in extracting key features from book blurbs, ensuring relevant recommendations. The system automates data preprocessing, saving time and effort, and is scalable to meet both small and enterprise-level needs.

Refrances

• Scikit-learn Documentation: https://scikit-learn.org/stable/

• Django Documentation: https://www.djangoproject.com/

• NumPy and pandas Official Guides: https://numpy.org/doc/ and https://pandas.pydata.org/docs/

• Matplotlib Documentation: https://matplotlib.org/stable/contents.html

• Data Cleaning Techniques: https://towardsdatascience.com/the-ultimate-guide-to-data cleaning-3969843991d4

• Feature Engineering Best Practices:
https://machinelearningmastery.com/discover feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/

• Natural Language Processing with Python: https://www.nltk.org/

• Building Recommender Systems with Python:
https://www.packtpub.com/product/building-recommender-systems-with python/9781788621759

• Introduction to Django REST Framework: https://www.django-rest-framework.org/

• HTML and CSS for Beginners:
https://www.freecodecamp.org/news/html-and-css tutorial-for-beginners/