Emotion-Powered Poem Search Engine

António Moura Coutinho
范才勇
2024403033

October 8, 2024

Abstract

This paper presents the development and implementation of a Poem Search Engine, designed to facilitate the retrieval of poems based on emotional content. The system integrates natural language processing (NLP) and information retrieval (IR) techniques to classify and index poems by their emotional vectors. This report covers the system's architecture, data processing methods, classification algorithms, and potential future improvements.

1 Introduction

The search for literary works, particularly poems, based on emotional content is a complex task. Traditional keyword-based search engines often fail to capture the nuanced emotional undertones of poems. This project aims to develop a search engine that categorizes and retrieves poems based on specific emotions, enhancing the discoverability of poetry that resonates with the reader's mood or interest.

Existing keyword-based search engines typically rely on the literal words in the text, which can overlook the deeper emotional resonance of poems. By contrast, our approach leverages advanced NLP techniques to better align search results with the emotional content users are seeking.

2 System Overview

The Poem Search Engine consists of several key components:

2.1 Backend

The backend is responsible for the core data processing tasks:

- **Data Retrieval and Cleaning:** Gathering and pre-processing a large corpus of poems, including removing duplicates, normalizing text, and ensuring consistent formatting.

- **Emotion Classification:** Analyzing each poem and assigning an emotion vector using a classifier trained on a dataset of annotated poems. The emotions considered include happiness, sadness, fear, disgust, anger, surprise, anticipation, trust, guilt, love, saudade, envy, bittersweetness, loneliness, and nostalgia.
- **Database Management:** Storing the processed poems and their corresponding emotion vectors in a CSV file for efficient retrieval.

2.2 Frontend

The frontend provides an interface for searching and displaying poems in 3D with a face to represent emotions. Users can search by additional criteria such as poet and title.

3 Data Processing and Classification

3.1 Data Retrieval and Cleaning

The system retrieves over 10,000 poems from various sources, with chinese poems of Tang dynasty poetry. The data cleaning process involves:

- Removing non-poetic text and metadata.
- Correcting formatting issues.
- Normalizing punctuation and spacing.

This script reads the raw poem data, removes unnecessary characters and whitespace, and ensures consistent formatting. It also prepares the emotion vectors by converting them from strings to lists of floats for further processing.

3.2 Emotion Classification

The classification of poems by emotion is achieved through the GPT-3 pretrained large language model [1]. The model uses NLP techniques to analyze the text and assign an emotion vector. Key steps include:

- **Tokenization:** Breaking down the text into individual words or phrases. This step is crucial for further analysis as it simplifies the text into manageable units. Although GPT-3 handles tokenization internally, additional tokenization may be done using Python's NLP libraries, such as NLTK or SpaCy, for any pre-processing or custom tokenization needs.
- **Feature Extraction:** GPT-3 inherently performs feature extraction by identifying key features that correlate with specific emotions within its neural network. However, additional feature extraction methods like TF-IDF or word embeddings (e.g., Word2Vec, GloVe) can be used for any supplementary processing if needed.

- **Model Training:** Since GPT-3 is a pre-trained model, no additional training is required. Instead, the model is utilized directly to predict the emotion vectors based on the input poems.
- **Emotion Classification using GPT-3:** The preprocessed text is fed into the GPT-3 model, which has been fine-tuned to predict emotion vectors based on the input poems. GPT-3 uses its extensive training on diverse datasets to identify and classify the emotions present in the poems.
- **Vector Assignment:** Each poem is assigned an emotion vector that represents the intensity and presence of various emotions, such as happiness, sadness, fear, disgust, anger, surprise, anticipation, trust, guilt, love, saudade, envy, bittersweetness, loneliness, and nostalgia.
- **Validation:** The performance of GPT-3 in classifying emotions is validated by comparing the predicted emotions with a set of manually annotated poems.

4 System Architecture

The Poem Search Engine's architecture consists of a backend and frontend, each responsible for different functionalities. The backend handles data processing and classification, while the frontend interacts with users and displays search results.

4.1 Backend

The backend comprises several modules:

- **Data Ingestion:** Collects raw poem data from various sources.
- **Preprocessing:** Cleans and normalizes the data.
- **Classification:** Applies the trained emotion classifier to each poem.
- **Database:** Stores the processed data and emotion vectors in a CSV file.

4.2 Frontend

The frontend features:

- **Search Interface:** Allows users to input search criteria and select desired emotions.
- **Results Display:** Shows a list of poems matching the search criteria, including metadata such as title and poet.

- **Interpret and Analyze Button:** A button to interpret and analyze poems, translating them into English to provide a deeper understanding of their emotional content.

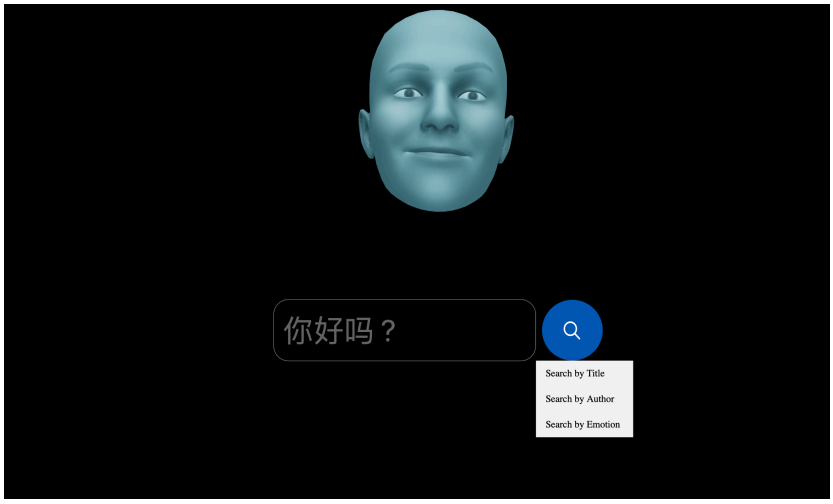

Figure 1: Search interface.

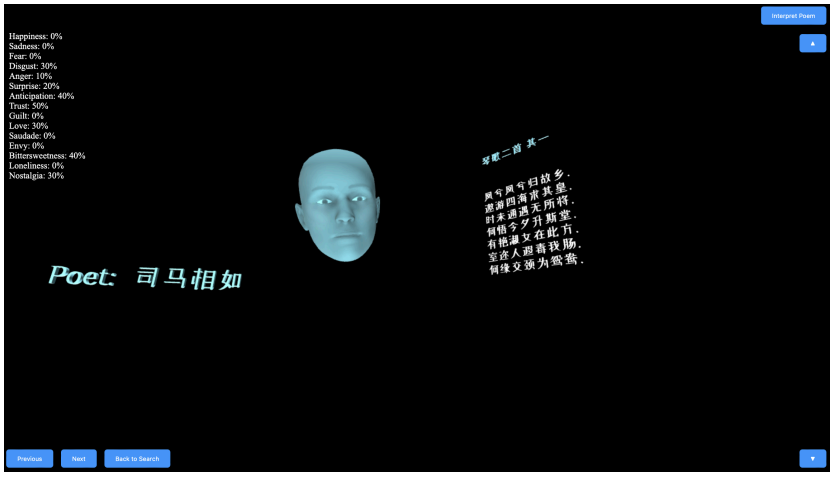

Figure 2: Search results.

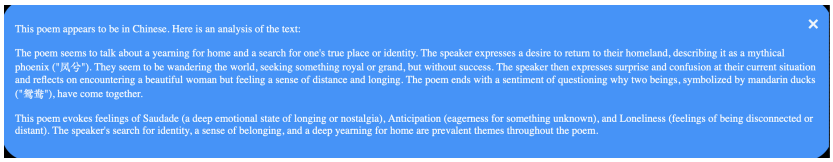

Figure 3: Interpretation and translation.

5 Implementation Details

5.1 Backend Implementation

The backend is implemented using Python and Flask[1], a lightweight web framework. Key components include:

- **Flask Application:** Handles HTTP requests and serves as the interface between the frontend and the backend.
- **Pandas:** Used for data manipulation and cleaning.
- **GPT-3 API:** Utilized for pre-analyzing and labeling poems with emotion vectors.
- **NLP Libraries:** Utilized for text processing and emotion classification.
- **Database:** Processed poems and their emotion vectors are stored in a CSV file.
- **Deployment:** The backend is deployed on Heroku for accessible and scalable cloud hosting.

5.2 Frontend Implementation

The frontend of the Poem Search Engine is developed using a combination of HTML, CSS, and JavaScript, ensuring an intuitive and responsive user interface:

- **HTML/CSS:** These technologies are used to create the structure and styling of the user interface. HTML provides the essential elements and layout of web pages, while CSS is used to enhance the visual appearance, making the interface user-friendly and aesthetically pleasing.
- **JavaScript:** JavaScript is employed to handle user interactions and provide dynamic content updates. It communicates with the backend API to retrieve and display search results based on user inputs. JavaScript also manages form submissions, input validation, and the dynamic updating of the interface without requiring page reloads.

[1] `https://flask.palletsprojects.com/en/3.0.x/`

- **Three.js:** This JavaScript library[2] is utilized for rendering 3D visualizations of poems and representing emotions through 3D faces. Three.js allows for the creation of interactive and visually appealing 3D graphics that enhance the user's experience when exploring poems based on their emotional content.
- **User Interface Components:**
  - **Search Interface:** A user-friendly search bar where users can input criteria such as emotions, poet names, or poem titles. The interface allows for selecting emotions from a predefined list to tailor the search results.
  - **Results Display:** Displays a list of poems that match the user's search criteria. Each result includes metadata such as the title, poet. The results are dynamically updated based on user inputs.
  - **Interpret and Analyze Button:** A feature that allows users to interpret and analyze selected poems. When activated, this button provides translations of the poems into English, offering a deeper understanding of their emotional content.

This frontend implementation ensures that users have an engaging and seamless experience when interacting with the Poem Search Engine, allowing them to explore and discover poems that resonate with their emotions effectively.

6 Search Functionalities

The Poem Search Engine provides three primary search functionalities:

- **Search by Emotion:** Users can input an emotion, which is transformed into an emotion vector. The system utilizes cosine similarity to match and return poems that closely align with the queried emotion. This feature allows users to discover poems that resonate with their current emotional state or desired mood.
- **Search by Poet:** Users can search for poems by specifying the poet's name. This functionality helps users find all poems written by a particular poet, enabling them to explore the works of their favorite poets or discover new ones.
- **Search by Title:** Users can search for poems by specifying the poem's title. This feature is particularly useful for users who are looking for a specific poem but may not remember the exact details, as partial titles can also be used to find matches using fuzzy search.

[2] `https://threejs.org/`

7 Results and Evaluation

The effectiveness of the Poem Search Engine is evaluated based on its accuracy in classifying poems by emotion and the user experience of the search interface. Key metrics include:

- **Classification Accuracy:** Measured by comparing the predicted emotions with a set of manually annotated poems.
- **User Feedback:** Collected through surveys and usability testing to assess the intuitiveness and usefulness of the search interface.

8 Future Improvements

The current system offers a robust foundation for emotion-based poem retrieval, but several enhancements can be made to further improve its effectiveness and user experience:

- **Enhanced Emotion Classifier:** Improve the accuracy of the emotion classifier by incorporating more advanced NLP techniques such as deep learning models (e.g., BERT [2], GPT-4 [3]) and by expanding the training dataset with a more diverse and extensive collection of annotated poems.
- **Additional Search Filters:** Introduce more refined search filters to enhance the search experience. Filters could include language, publication date, poem length, and even thematic categories, allowing users to narrow down their searches more effectively.
- **User Interface Enhancements:** Redesign the user interface to be more intuitive and aesthetically pleasing. This includes improving the layout, navigation, and overall visual design to make the platform more user-friendly and engaging.
- **Backend Optimization:** Optimize backend processes to improve the system's speed and efficiency. This involves refining data processing algorithms, improving database query performance, and enhancing server-side optimizations to reduce search and retrieval times.
- **Educational Integration:** Develop and implement a gamified educational tool based on the poem search engine for use in schools. This tool could help students learn about poetry and emotions in an interactive and engaging way, fostering a deeper appreciation for literature while enhancing their emotional literacy. Collaborating with teachers can further refine the classification of data, ensuring the tool's effectiveness in an educational setting.

9 Conclusion

The Poem Search Engine represents a significant advancement in the field of literary search engines, offering a novel approach to discovering poetry based on emotional content. By leveraging advanced natural language processing (NLP) and information retrieval (IR) techniques, the system effectively categorizes and retrieves poems that resonate with the emotional states of users.

The implementation of a robust backend, responsible for data ingestion, preprocessing, emotion classification, and database management, ensures the reliability and efficiency of the search engine. The frontend interface enhances user experience by providing an intuitive and interactive platform for exploring poems through emotional filters, poet names, and poem titles.

While the current system demonstrates considerable promise, there are several areas for potential improvement. Enhancing the accuracy of the emotion classifier through more sophisticated NLP techniques and larger training datasets, introducing additional search filters, and optimizing the user interface and backend processes are key directions for future work.

In conclusion, the Emotion-Powered Poem Search Engine not only broadens the accessibility and appreciation of poetry but also provides a meaningful way for users to connect with literary works on an emotional level. The continued development and refinement of this system have the potential to further enrich the user experience and solidify its place as an essential tool for literary exploration.

References

[1] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[2] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186.

[3] OpenAI. "GPT-4 Technical Report". In: *ArXiv* abs/2303.08774 (2023).