SI 650 Project Report

Ayush Shah & Shloki Jha

Introduction

The aim of this project is to develop a lyrics-based search engine capable of retrieving songs based on partial lyrics input. Existing search engines often struggle with incomplete or inaccurate lyrics, resulting in poor user experiences. To address these challenges, our system incorporates both baseline rankers (BM25, TF-IDF, etc.) and advanced models like Siamese BERT and Latent Semantic Analysis (LSA). By leveraging relevance scores manually annotated for realistic evaluation, this project demonstrates the strengths and limitations of combining statistical and semantic approaches in the context of lyrics-based information retrieval. Solving this problem has significant value for improving music search engines, making them more robust and user-friendly.

Data

The dataset used for this project was sourced from the Genius Lyrics Dataset on Hugging Face, which contains over 180,000 song lyrics along with metadata such as artist names, album names, and release dates. For manageability, a subset of 50,000 documents was selected for this project. From this subset, 20 queries were uniformly generated, each retrieving 50 documents, resulting in a total of 1,000 query-document pairs. These pairs were manually annotated on a relevance scale of 1–5, where scores of 1–2 indicated low relevance, 3 represented moderate relevance, and 4–5 signified high relevance.

To prepare the data for analysis, several preprocessing steps were undertaken. Missing or irrelevant data entries were removed to ensure a clean dataset. Lyrics were tokenized using the RegexTokenizer to maintain consistency in text processing, followed by stopword filtering to reduce noise and improve the quality of features. Metadata such as genres were converted from raw strings to structured lists, ensuring better alignment with the project's preprocessing and analysis needs. These steps collectively ensured that the dataset was ready for effective indexing and retrieval.

Relevance_Score_Distribution

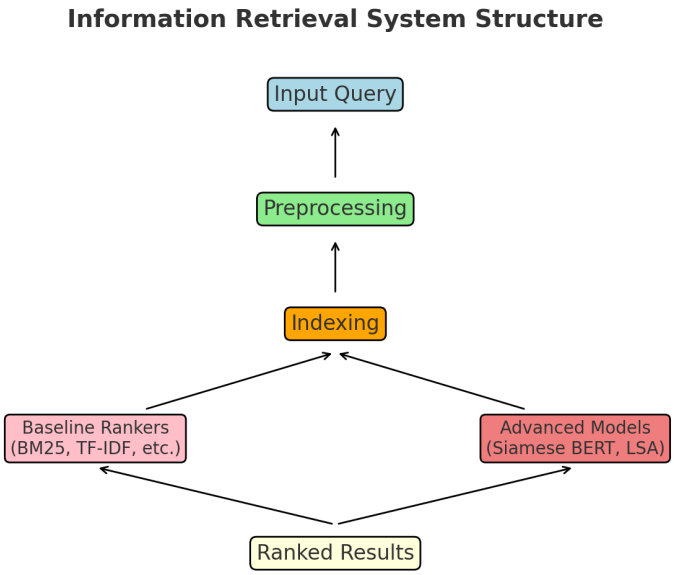
Relevance Score	Count
1	122
2	677
3	164
4	13
5	24

Related Work

1. **SongWords**: Baur et al. developed a system using self-organizing maps for exploring music collections through lyrics. This study highlighted the potential of lyrics for organizing music collections and inspired our focus on lyrics as a central feature.
2. **Phonetic Lyric Search**: Xu et al. proposed a method using a phonetic confusion matrix and dynamic programming for robust lyric search. This influenced the design of semantic similarity components in our project.
3. **Mood-Based Classification**: Dang and Shirai used sentiment analysis and machine learning to classify songs by mood. Their findings supported the importance of contextual features in retrieval tasks.
4. **Approximate Phrase Searching**: Patterson and Watters introduced proximity-based phrase matching for inexact lyrics, a feature considered for enhancing search robustness in our system.
5. **Genre-Based Retrieval**: Although we dropped genre-based recommendations, related studies provided insights into the complexities of integrating metadata with lyric-based retrieval.

Methods

This project involved implementing and evaluating several ranking methods, ranging from traditional statistical rankers to advanced machine learning models. These methods were chosen to address the challenges posed by lyric-based retrieval, such as incomplete user inputs and the semantic richness of lyrics.



Baseline Rankers:
The baseline rankers included BM25, TF-IDF, WordCountCosineSimilarity, and Pivoted Normalization. BM25, a cornerstone in information retrieval, was selected for its balance between term frequency and inverse document frequency, providing robust rankings for keyword-based queries. TF-IDF offered a simpler yet effective way to assess document relevance by computing term importance across the dataset. WordCountCosineSimilarity was chosen for its interpretability, measuring similarity between the query and document word vectors. Pivoted Normalization was included as a baseline to adjust document length normalization dynamically, enabling better handling of length variations in song lyrics. These baseline rankers served as benchmarks for evaluating the effectiveness of more complex models.

Advanced Models:
To go beyond the limitations of statistical rankers, two advanced models were implemented: Siamese BERT and Latent Semantic Analysis (LSA).

Siamese BERT was chosen for its ability to capture semantic relationships in text, which is crucial for lyric-based search where users often input paraphrased or incomplete queries. This model employs pre-trained embeddings from the all-MiniLM-L6-v2 architecture to represent both queries and documents in a high-dimensional semantic space. By precomputing embeddings for all documents, the system efficiently computes cosine similarity between a query and the document embeddings at runtime. Despite its computational cost, Siamese BERT demonstrated remarkable performance, accurately retrieving lyrics with semantic similarities. However, the high memory requirements and computational overhead made it less scalable for larger datasets.

Latent Semantic Analysis (LSA) offered a balance between complexity and efficiency. LSA transforms lyrics into a lower-dimensional semantic space by first generating TF-IDF vectors and then applying dimensionality reduction using Truncated Singular Value Decomposition (SVD). In this reduced space, cosine similarity measures the relevance between queries and documents. LSA was particularly effective in capturing latent patterns and conceptual similarities in lyrics, making it a suitable alternative to more resource-intensive neural models. However, its reliance on dimensionality reduction occasionally resulted in oversimplification, losing some of the nuances in the data.

Evaluation Metrics:
The effectiveness of all rankers was evaluated using Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG). Weighted MAP was used to incorporate graded relevance levels, ensuring higher relevance scores had more impact on the evaluation. Normalized NDCG accounted for the position of relevant documents in the rankings, emphasizing early retrieval of highly relevant lyrics.

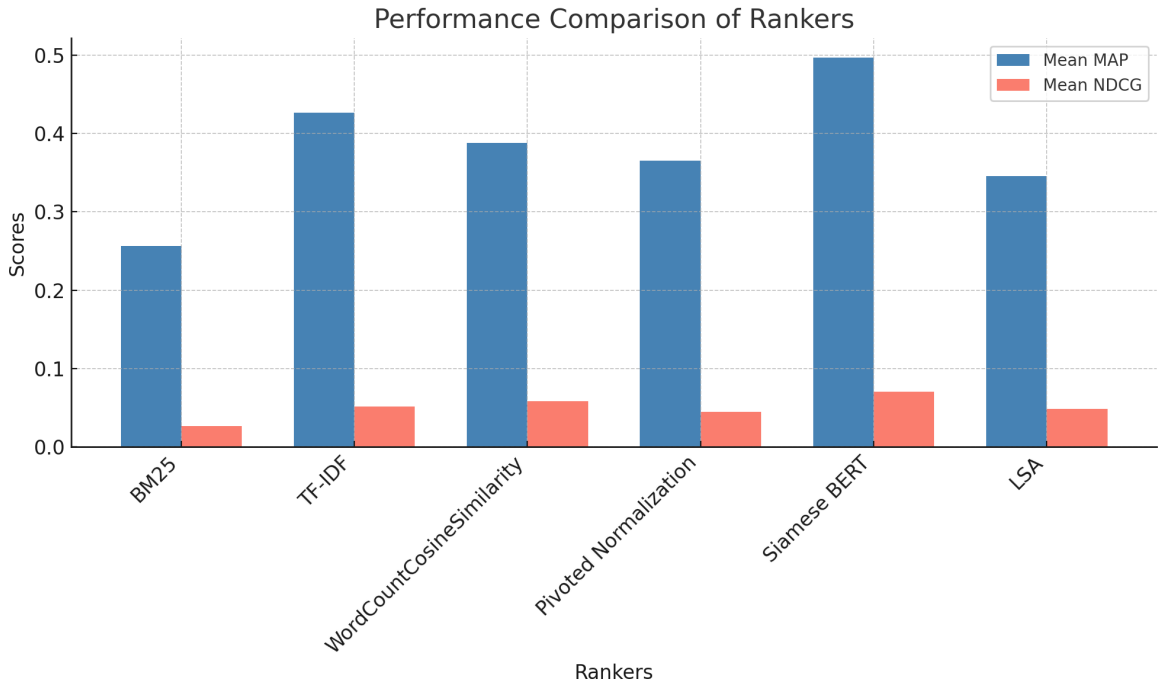
Workflow:
The system's workflow began with preprocessing the lyrics dataset, where tokenization and stopword filtering were applied using the RegexTokenizer. The processed data was indexed using a BasicInvertedIndex, allowing efficient retrieval of documents. Ranking was performed using both baseline and advanced models, and the results were evaluated using the annotated relevance scores to calculate MAP and NDCG metrics.

This combination of baseline and advanced methods provided a comprehensive evaluation framework, enabling a detailed analysis of the strengths and limitations of each approach in addressing the challenges of lyric-based retrieval.

Evaluation and Results

The implemented information retrieval system was evaluated using both baseline and advanced rankers on a dataset of 1,000 manually annotated query-document pairs. The evaluation metrics employed were Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG), selected for their effectiveness in assessing the quality of ranked document lists.

The baseline rankers included BM25, TF-IDF, WordCountCosineSimilarity, and Pivoted Normalization. Among these, TF-IDF achieved the highest MAP score (0.4268), followed by WordCountCosineSimilarity (0.3883). BM25, a commonly used statistical ranker, performed less effectively in this setup, with a MAP of 0.2566 and an NDCG of 0.0269, potentially due to the nature of the lyric dataset and the impact of term distribution. Pivoted Normalization ranked similarly to WordCountCosineSimilarity but with a slightly lower MAP of 0.3657. Overall, the baselines provided reliable benchmarks for comparing more advanced methods.



The advanced rankers, Siamese BERT and Latent Semantic Analysis (LSA), demonstrated improved performance over the baselines. Siamese BERT achieved the highest scores among all rankers, with a MAP of 0.4972 and an NDCG of 0.0704. This reflects its ability to capture semantic nuances and effectively handle paraphrased or incomplete queries. LSA, although less effective than Siamese BERT, outperformed most baselines, achieving a MAP of 0.3457 and an NDCG of 0.0488. Its performance highlights the value of capturing latent relationships in the text while remaining computationally efficient compared to deep learning models.

The results underscore the importance of incorporating semantic understanding into information retrieval systems. While advanced methods such as Siamese BERT clearly excel in handling complex queries, the simplicity and efficiency of baseline rankers like TF-IDF and WordCountCosineSimilarity make them viable options for resource-constrained environments.

This comprehensive evaluation provides a strong foundation for future work, demonstrating the effectiveness of both traditional and modern IR techniques in lyric-based retrieval tasks. The insights gained here pave the way for further optimization and scalability enhancements in future work.

Discussion

The evaluation highlights key insights into the performance of different rankers. Siamese BERT excelled in handling semantic similarity, proving highly effective for partial or paraphrased queries. Its ability to capture contextual nuances significantly improved ranking accuracy. LSA, while slightly less accurate, provided competitive performance by effectively capturing latent relationships in lyrics through dimensionality reduction.

However, the computational cost of Siamese BERT limits its scalability for large datasets, making it challenging for deployment in resource-constrained environments. LSA, though efficient, may overlook critical details in sparse datasets due to its reliance on dimensionality reduction.

Future work could explore integrating phonetic similarity to address user errors in input, enhancing retrieval robustness. Additionally, experimenting with hybrid rankers that combine the efficiency of statistical methods with the semantic understanding of neural approaches offers a promising direction for balancing performance and scalability.

Conclusion

This project successfully developed a robust lyrics-based search engine, combining baseline rankers like BM25 and TF-IDF with advanced models such as Siamese BERT and Latent Semantic Analysis (LSA). Through rigorous evaluation using manually annotated datasets, the system demonstrated the effectiveness of semantic models in handling paraphrased and partial queries while highlighting the efficiency of statistical methods for simpler applications. The findings underscore the trade-offs between accuracy and computational scalability, providing valuable insights for future work. Potential directions include integrating phonetic similarity and

exploring hybrid approaches to balance precision and performance. This project showcases the practical applications of information retrieval techniques and lays the groundwork for further innovation in domain-specific search systems.

Other Things We Tried

As part of the exploration, a bi-encoder model was implemented. This approach encodes queries and documents independently into vector representations and computes cosine similarity for ranking. While this model was expected to balance efficiency and accuracy, it did not perform well on the manually annotated dataset. Likely reasons include insufficient fine-tuning of the pre-trained model for lyric-based tasks and limitations in capturing complex semantic nuances. Despite its challenges, significant time was spent on this implementation as it was a promising direction initially. Including the bi-encoder experiment highlights the project's exploratory nature and the iterative learning process involved in refining the system.

What You Would Have Done Differently or Next

Given more time and resources, one of the primary areas for improvement would be developing a working interface for the search engine. This would enable real-time user interactions, allowing us to test the system's usability and gather feedback for further refinements. Additionally, experimenting with more advanced models, such as transformer-based cross-encoders or domain-specific embeddings, could enhance the retrieval accuracy and robustness of the system. These models, while computationally intensive, may better handle the nuances of lyrics-based queries. Combining these with phonetic similarity features and hybrid rankers could create a more refined and user-friendly lyrics-based search engine. Exploring these aspects in the future would help improve both the performance and practicality of the system.

Team Work Distribution

The project tasks were distributed as follows:

- **Ayush Shah**: Focused on data preprocessing, implementing baseline rankers (BM25, TF-IDF, WordCountCosineSimilarity, and Pivoted Normalization), and setting up the evaluation framework for system performance assessment.
- **Shloki Jha**: Led the implementation of advanced rankers, including Siamese BERT and Latent Semantic Analysis (LSA), and conducted experimental analyses to evaluate the effectiveness of these models against the baseline methods.