Analyzing and Classifying Harry Potter Data

Abhishek Bargujar, Martin Kafka
Information Retrieval

Winter Semester 2024/2025

Abstract

This report presents a project focused on the development of an information retrieval system using datasets related to the Harry Potter universe. The objective of the project was to develop a system capable of efficiently retrieving textual information. Two approaches were explored: traditional methods involving manual preprocessing and query expansion, and modern methods utilizing transformers and embeddings for advanced retrieval. The efficacy of the system was evaluated using metrics such as Precision@K and Mean Reciprocal Rank (MRR), with a demonstration through an interactive application. The subsequent sections of this report will discuss potential future improvements and extensions.

1 Introduction

1.1 Motivation

The decision to focus on the Harry Potter topic was made in recognition of the mutual interest held by both of us in the series's books and movies. The thematic scope of the Harry Potter series, characterized by its extensive textual content and the diversity of its fictional universe, including characters, spells, and potions, renders it a compelling case study for the investigation of information retrieval methodologies.

Additionally, the availability of structured datasets on Kaggle offered a sufficient amount of data, including corpora from the first three movies, as well as detailed information about spells and potions used throughout the series. This made it possible to build a meaningful retrieval system and classification models with practical applications.

1.2 Objective

The main goal of this project is to develop an efficient information retrieval system by applying various methods learned during the course. Specifically, the project aims to:

- Load data from multiple Harry Potter-related datasets, including dialogues, spells, and potions. This involves handling missing values and standardizing the text for consistency.
- Build an inverted index for efficient query retrieval and implement TF-IDF to rank documents by their relevance to user queries.
- Implement cosine similarity and the Binary Independence Model (BIM) to evaluate query-document relevance for improved retrieval accuracy.
- Explore transformer-based embeddings and semantic search for improved retrieval capabilities.
- Develop an interactive application using Flask, providing a user-friendly interface for querying and retrieving the most contextually relevant sentences from the corpus based on the user's input.
- Evaluate the performance of the system using metrics such as Precision@K and Mean Reciprocal Rank (MRR).

2 Methodology

2.1 Data Collection and Cleaning

The data for this project was collected from the Kaggle platform, which provided structured datasets related to the Harry Potter universe. Specifically, the datasets included dialogues from the first three movies, detailed descriptions of spells, and information about potions. All these data are in CSV format.

2.1.1 Data Cleaning

Several preprocessing steps were performed to clean the data:

- **Standardization of column names**: Column names were standardized by converting them to lowercase, replacing spaces with underscores, and removing special characters to ensure consistency across datasets.
- **Handling missing values**: Missing values were handled based on the type of data:
  - Categorical columns (e.g., character names) were filled with the value `"Unknown"`.
  - Numerical columns were filled with zeros.
- **Text normalization**: The text data was cleaned by removing special characters, extra spaces, and converting all text to lowercase. This was essential for consistent text processing in later stages.

The result of these steps was a clean and standardized dataset, ready for further preprocessing and analysis.

2.2 Text Preprocessing for Information Retrieval

Text preprocessing is a crucial step in developing an efficient information retrieval system, as it ensures consistency and improves the quality of retrieval. Several preprocessing techniques were applied using the `NLTK` and `re` libraries in Python:

- **Tokenization**: Sentences and dialogues were split into individual words (tokens) using NLTK's tokenizer.
- **Stopword Removal**: In order to prioritize terms that are more meaningful, common English stopwords were removed. These stopwords were obtained from the NLTK library, which provides a comprehensive list of common stopwords for multiple languages.
- **Stemming and Lemmatization**: Stemming was performed using the Porter stemmer to reduce words to their root form. Additionally, lemmatization was applied to convert words to their base form while preserving grammatical context.
- **Text Normalization**: All text was converted to lowercase, and special characters, extra spaces, and punctuation were removed using regular expressions.

2.3 Query Expansion

Two distinct approaches were explored for query expansion and document retrieval:

1. **Traditional Approach:** In this approach, bigrams were generated from tokenized sentences and indexed using an inverted index, and queries were expanded by suggesting additional terms based on their frequency in the corpus. This approach functions similarly to a query autocomplete system, providing relevant word suggestions rather than full document retrieval. Document relevance was determined using Cosine Similarity and the

Binary Independence Model (BIM), ranking documents based on their likelihood of containing terms related to the query. While this approach was found to be efficient for simple queries, it lacked the ability to capture semantic relationships or provide context-aware suggestions.

2. **Modern Approach:** The modern approach utilized transformer-based embeddings generated using the **Sentence Transformer** model all-MiniLM-L6-v2 from the HuggingFace library. The model under consideration creates semantic representations of both queries and documents in a high-dimensional vector space. Rather than suggesting individual terms, this approach retrieves the most contextually relevant sentences from the corpus by computing the cosine similarity between query and document embeddings. This enables precise semantic search, making it highly effective for complex and ambiguous queries.

The fundamental distinction between these approaches lies in their scope and depth. The Traditional Approach offers relevant word suggestions, functioning as a query autocomplete tool. Conversely, the Modern Approach identifies and retrieves entire sentences that best match the query, thereby providing a more nuanced and semantically aware search experience.

2.3.1 Methodology for Query Expansion

- Bigrams were generated from tokenized sentences using sliding windows.
- An inverted index for bigrams was constructed to map bigrams to the corresponding documents.
- When a query was entered, bigrams related to the query terms were identified, and the most frequent associated terms were suggested for expansion.
- **Cosine Similarity**: Used to measure the similarity between the query vector and document vectors, ensuring the retrieval of the most relevant documents.
- **Binary Independence Model (BIM)**: Applied for probabilistic ranking of documents by estimating the likelihood of relevance based on term presence in documents.

2.3.2 Benefits of Query Expansion

The primary benefit of query expansion is improved recall, as it enables the retrieval system to consider documents containing related terms that may not be explicitly mentioned in the original query. This process leads to a more comprehensive set of results and enhances the overall user experience.

3 Results and Discussion

3.1 Information Retrieval Results

The information retrieval system was evaluated using two key metrics: Precision@K and Mean Reciprocal Rank (MRR). These metrics were applied to the traditional approach using cosine similarity.

The traditional approach performed reasonably well for straightforward queries, achieving a Precision@1 of 0.25. However, its performance declined significantly as K increased, highlighting its limitations in handling more complex queries. Additionally, the bigram-based indexing approach for query expansion occasionally failed to return results for certain keywords, likely due to the sparse nature of the dataset and class imbalance.

In contrast, the transformer-based approach retrieved entire sentences that were semantically relevant to the query. By leveraging contextual embeddings, it demonstrated superior performance for complex and ambiguous queries, addressing the shortcomings of the traditional approach.

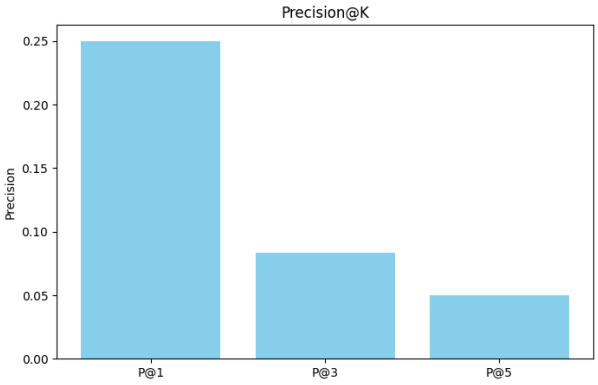

Figure 1: Precision@K

3.2 Comparison of Approaches

- **Traditional Approach:** This approach was efficient for simple queries and provided suggestions similar to query autocomplete. However, the bigram-based indexing sometimes failed to return any results for certain keywords, likely due to the nature of the dataset and the imbalance in the distribution of terms. Precision@K metrics showed a sharp decline as more results were considered, indicating a limited ability to rank documents effectively for complex queries.

- **Modern Approach:** The transformer-based retrieval method retrieved entire sentences that best matched the query in a semantic sense. It performed consistently well even for ambiguous or context-dependent queries by leveraging embeddings from the all-MiniLM-L6-v2 model. This approach addressed the limitations of the traditional method and provided a robust solution for semantic information retrieval. See the Example Interaction section.

3.3 Discussion

The information retrieval system demonstrated distinct strengths and weaknesses across the two approaches. The traditional approach, based on cosine similarity and bigram indexing, showed reasonable performance for simple queries but struggled with complex queries and semantic nuances. Its reliance on bigram frequency often led to cases where no suggestions or results were returned for certain keywords, particularly due to the sparsity of the dataset and class imbalance.

On the other hand, the transformer-based approach excelled in retrieving semantically relevant sentences, even for ambiguous or contextually rich queries. By leveraging embeddings, it overcame the limitations of bigram-based expansion and provided a more meaningful and accurate retrieval experience.

The comparison highlights the importance of considering the nature of the dataset and the query complexity when selecting a retrieval method. While the traditional approach offers simplicity and efficiency, the modern transformer-based approach is better suited for complex information retrieval tasks.

4 Interactive Application Development

The interactive application was redesigned to utilize the transformer-based retrieval approach, enabling users to input queries and retrieve the most contextually relevant sentences from the document corpus. The application includes the following key components:

- **Input Query Processing:** Users enter queries into a simple HTML interface. These queries are converted into embeddings using the all-MiniLM-L6-v2 transformer model.
- **Sentence Ranking and Retrieval:** The application computes cosine similarity between the query embedding and document embeddings to rank sentences based on their relevance to the input query.
- **Query Suggestions:** In cases where the query is ambiguous, related terms and suggestions are displayed to guide the user toward more precise searches.

The application was developed using `Flask`, with the backend integrating the transformer model for real-time embedding generation and sentence retrieval. This architecture ensures that users receive the most relevant sentence in response to their query, enhancing the overall retrieval experience.

4.1 Example Interaction

For example, when a user enters the query `"McGonagall"`, the application retrieves and ranks the most contextually relevant sentences from the corpus. The results include:

- Professor McGonagall! (Score: 0.83)
- McGonagall gave it to me first term. (Score: 0.70)
- He won't keep it. He'll turn it over to Professor McGonagall. Aren't you? (Score: 0.54)
- Voldemort. (Score: 0.41)
- Peter Pettigrew. (Score: 0.38)

These results demonstrate the application's ability to rank sentences based on semantic relevance to the input query, with higher scores indicating greater contextual similarity.

5 Limitations and Future Work

5.1 Limitations

While the developed system yielded beneficial outcomes, several limitations were identified during the course of the project:

- **Limited Dataset Scope**: The datasets primarily included dialogues from the first three Harry Potter films, along with spells and potions. The limited scope of the data may have impacted the system's ability to handle diverse or highly specific queries.
- **Bigram-Based Query Expansion Limitations**: The traditional query expansion method, relying exclusively on bigram analysis, occasionally failed to return results for certain keywords, especially when the terms were rare or contextually ambiguous. This limitation was inherent to the dataset's structure and distribution of terms.
- **Transformer Dependence**: While the transformer-based approach significantly improved retrieval accuracy, its computational requirements could pose scalability challenges for larger datasets or real-time applications.

5.2 Future Work

The following potential extensions and improvements to the project can be considered:

- **Expanding the Dataset**: Incorporating dialogues from all Harry Potter movies and books, as well as additional metadata (e.g., character profiles, locations, and events), would allow for more comprehensive retrieval and improve the system's robustness.
- **Enhanced Query Expansion**: Further refining the query expansion mechanism by integrating advanced techniques such as contextual embeddings or knowledge graphs could address the shortcomings of bigram-based expansion.
- **Scalability Improvements**: Utilizing advanced transformer models like BERT could further enhance the system's ability to handle larger datasets and provide more accurate retrieval results.
- **Application Usability Enhancements**: Developing a more interactive and feature-rich user interface, such as visualizing query results or providing feedback on suggested terms, would improve the overall user experience.

6 Conclusion

This project developed an information retrieval system with two distinct approaches: a traditional method based on manual preprocessing and query expansion and a modern method utilizing transformer-based embeddings. The transformer-based approach demonstrated superior performance in semantic retrieval tasks, while the traditional approach provided efficient results for simpler queries. An interactive web application was developed to showcase the system's capabilities, highlighting the value of integrating retrieval techniques with advanced models for improved user experience.

7 References

Reference to GitLab with all necessary data, output screens, code and application.