Report: Enhancing Recommendation Systems with NLP

By Alex Szpakiewicz and Léonard Roussard

Project Overview

This project aimed to develop a recommendation system capable of identifying the most semantically similar hotel based on user reviews. The objective was to surpass BM25 without preprocessing—a foundational information retrieval model. By exploring hybrid approaches, transformer-based architectures, and specialized models, the goal was to improve both accuracy and efficiency while addressing BM25's inherent limitations.

Introduction

The dataset used was the **TripAdvisor Hotel Review dataset**, which includes textual reviews and ratings for seven aspects: service, cleanliness, overall experience, value, location, sleep quality, and rooms. To prepare the dataset:

1. **Filtering**: Reviews missing ratings for any of the seven aspects were removed.
2. **Aggregation**: Reviews were grouped by the `offering_id` attribute, consolidating all textual data for each hotel.
3. **Averaging**: Aspect ratings were averaged for each hotel to create a unified metric for evaluation.

Using my MacBook Pro with an M3 Pro CPU and GPU, I was able to process the entire dataset efficiently. Unlike in cloud environments like Colab, where performance limitations often restrict data usage, my setup enabled comprehensive experimentation without compromising speed.

BM25 Baseline Evaluation

What is BM25?

BM25 is a ranking function based on keyword matching between a query and documents. It evaluates:

1. **Term Frequency (TF)**: Importance of a word within a document.
2. **Inverse Document Frequency (IDF)**: Rarity of the word across the entire corpus.
3. **Length Normalization**: Preference for shorter documents when scores are otherwise identical.

BM25 is simple yet effective, making it a common baseline for information retrieval tasks.

Results

BM25 was tested in two scenarios:

1. **Without Preprocessing (Baseline for Comparison)**: Raw reviews were used.
    - Average Scoring Time per Query: **10.55 seconds**
    - Total Time: **1055.26 seconds**

    - MSE: **0.5476**

2. **With Preprocessing**: Tokenization, stop-word removal, and text cleaning were applied as a custom improvement.

    - Average Scoring Time per Query: **3.73 seconds**
    - Total Time: **373.23 seconds**
    - MSE: **0.4910**

While BM25 with preprocessing demonstrated notable improvements, the primary baseline for comparison in this project was BM25 without preprocessing. Notably, all other models were implemented on the cleaned dataset, showcasing BM25's robustness against more complex models.

Approach to Model Exploration

The iterative approach started with BM25 and sought to address its limitations. BM25 without preprocessing, while effective, had a high MSE and slow processing speed. My first improvement involved combining BM25 with semantic embeddings to better capture meaning. However, the hybrid model still relied on BM25's CPU-bound processing, limiting speed.

Subsequent efforts focused on models capable of leveraging GPU acceleration to significantly reduce query time. These included transformer-based architectures like MPNet and sentence comparison models like SimSec. Finally, I tested the state-of-the-art **MXBI Colbert**, which combines advanced semantic understanding with efficient retrieval mechanisms, achieving promising results.

Process and Model Analysis

Hybrid Model

The hybrid model combined BM25's keyword-based scoring with semantic embeddings from a pre-trained model. This approach aimed to enhance semantic understanding while retaining BM25's precision in lexical matching.

- **Performance**: MSE of **0.4910**
- **Efficiency**: Average query time of **4.01 seconds**; total processing time of **401.19 seconds**
- **Limitations**: The reliance on BM25 meant the hybrid model was CPU-bound, significantly impacting speed.
- **Analysis**: While this model demonstrated meaningful improvement in accuracy, its speed limitations made it impractical for large-scale applications.

Transformer-Based Model: MPNet

MPNet is a transformer-based architecture designed for contextual embedding generation, capturing nuanced relationships between words.

- **Performance**: MSE of **0.4637**
- **Efficiency**: Leveraged GPU acceleration, achieving an average query time of **0.05 seconds**
- **Analysis**: MPNet offered excellent speed advantages due to its GPU support and showed good accuracy improvements over the baseline.

SimCSE Model

The SimCSE model focuses on sentence similarity using contrastive learning techniques.

- **Performance**: MSE of **0.4162**, best among all tested models
- **Efficiency**: Very efficient, with an average query time of **0.06 seconds**
- **Analysis**: SimCSE demonstrated both superior accuracy and excellent efficiency, making it the top performer in our evaluation.

MXBAI Models

MXBAI (MixedBread AI) models are specialized neural architectures designed for efficient information retrieval and semantic search. They offer two main variants:

MXBAI Embed

MXBAI Embed is a bi-encoder model that generates dense vector representations of text. It:

- Uses a BERT-based architecture optimized for semantic similarity
- Processes queries and documents independently for efficient retrieval
- Employs contrastive learning techniques during training
- Optimizes for both accuracy and inference speed

Results:

- **MSE**: **0.5167**
- **Average Query Time**: **0.13 seconds**
- **Total Processing Time**: **13.43 seconds**

MXBAI ColBERT

MXBAI ColBERT is an advanced neural retrieval model that combines the benefits of dense retrieval with token-level interactions. Key features include:

- Late interaction architecture allowing fine-grained matching
- Token-level representations for more precise similarity computation
- Efficient implementation of the ColBERT retrieval mechanism
- Balance between computational efficiency and retrieval accuracy

Results:

- **MSE**: **0.4954**
- **Average Query Time**: **0.13 seconds**
- **Total Processing Time**: **13.03 seconds**

SimCSE Model: Architecture and Implementation

As our best performing model, SimCSE (Simple Contrastive Learning of Sentence Embeddings) deserves special attention. Here's a detailed look at its architecture and implementation:

Architecture Overview

SimCSE employs a sophisticated contrastive learning framework:

1. **Base Architecture**: Built on RoBERTa-large, fine-tuned specifically for semantic similarity
2. **Dual Encoding**: Processes both queries and documents through the same encoder
3. **Pooling Strategy**: Uses [CLS] token representation with additional attention-weighted pooling
4. **Normalization**: L2 normalization on the final embeddings

Training Methodology

SimCSE's strength comes from its training approach:

1. **Unsupervised Contrastive Learning**:
    - Uses dropout as positive examples
    - Treats other sentences in batch as hard negatives
2. **Temperature Scaling**: Carefully tuned temperature parameter for similarity scores
3. **Data Augmentation**: Implicit augmentation through different dropout patterns

Implementation Details

The model implementation involves several key components:

1. **Tokenization**: Maximum sequence length of 512 tokens
2. **Embedding Generation**:
    - Forward pass through transformer layers
    - Attention-pooling over token embeddings
3. **Similarity Computation**:
    - Cosine similarity between normalized embeddings
    - Optional temperature scaling for final scores

Why It Performs Best

SimCSE achieves superior performance (MSE: 0.4162) due to:

1. **Effective Semantic Understanding**: Captures nuanced meanings through contrastive learning
2. **Efficient Processing**: Fast inference time (0.06s per query)
3. **Robust Representations**: Less sensitive to surface-level variations
4. **Balance**: Optimal trade-off between computational efficiency and accuracy

Comparative Results

The models, sorted by MSE, are summarized below:

Model	Avg Scoring Time (s)	Total Time (s)	MSE
SimCSE	0.06	6.16	0.4162
MPNet	0.05	5.09	0.4637
BM25 (with preprocess)	3.73	373.23	0.4910
Hybrid Model	4.01	401.19	0.4910

Model	Avg Scoring Time (s)	Total Time (s)	MSE
MXBAI ColBERT	0.13	13.03	0.4954
MXBAI Embed	0.13	13.43	0.5167
BM25 (no preprocess)	10.55	1055.26	0.5476

Conclusion

The evaluation revealed several interesting findings:

- SimCSE emerged as the best performing model, achieving both superior accuracy (MSE: 0.4162) and excellent efficiency.
- MPNet showed strong performance with the second-best accuracy (MSE: 0.4637) and very fast processing times.
- BM25 with preprocessing and the hybrid model tied for third place in accuracy (MSE: 0.4910).
- The MXBAI models, while efficient, didn't outperform the other approaches in terms of accuracy.
- BM25 without preprocessing served as a baseline but was outperformed by most other approaches.

Insights and Learning Outcomes

This project demonstrated the effectiveness of modern transformer-based approaches, particularly SimCSE and MPNet, which significantly outperformed traditional methods like BM25. The results highlight how recent advances in NLP can provide both better accuracy and faster processing times when properly implemented. The success of SimCSE in particular shows the value of contrastive learning approaches in semantic similarity tasks.