# Machine Learning for Natural Language Processing
# Project 1: TripAdvisor Recommendation Challenge

Michele PULVIRENTI, Marco RIVA

November 2024

# Contents

# 1 Introduction

This project involves the task of building a recommendation system based on TripAdvisor reviews, beginning with the implementation of a BM25 baseline and extending to the exploration of alternative or complementary approaches to improve performance.
To improve data quality, preprocessing steps were applied to ensure consistency and relevance in the input data, with attention to removing noise while preserving the semantic integrity of the reviews.

Additionally, various models and techniques were systematically tested to evaluate their effectiveness in capturing the nuances of user reviews. By exploring and combining different methodologies, the aim was to identify the optimal solution or hybrid approach capable of outperforming BM25, as measured by the evaluation protocol specified in the project.

We saved intermediate steps and data using `parquet` and `pickle` to not recompute everytime them.

# 2 Data Preprocessing

## 2.1 Preparation

First the dataset was prepared as described in the project's paper, removing all the columns different from those relevant for the ratings (and obviously the reviews' text), and keeping all those reviews having (only) the relevant ratings criterias.
It was also verified that there were no missing values after this data clearance.

## 2.2 Language

During the inspection of the dataset, it was observed that not all reviews were provided in the same language. While the majority of reviews appeared to be in English, many other languages were also present (e.g., French, Spanish, German, Japanese). An assessment indicated that more than 90% of the reviews were in English, leading to the decision to exclude reviews in other languages to ensure a consistent data source.

For language detection, an initial attempt was made using the simplest possible approach: identify a language checking if the text contains relevant characters of that language. However, it was quickly determined that achieving accurate results with this method was nearly impossible. Consequently, alternative approaches involving language detection libraries were explored to annotate each review with its language and retain only the English ones.

Specifically, two libraries were tested:

- `langdetect`
- `fast-langdetect`

The second library was selected due to its faster processing time while maintaining comparable accuracy to the first.

## 2.3 Transformation

In order to feed the `BM25` baseline model, on top of the preparation already performed, these transformations were applied to the dataset, using the `NLTK` library:

- **Lowercasing:** converting all characters in the text to lowercase.

- **Tokenization:** splitting the text into individual words (tokens).
- **Removing Punctuation and Stopwords:** filtering out tokens that are not alphanumeric and removes common stopwords.
- **Lemmatization:** converting words to their base or root form using a lemmatizer.

# 3 Model Testing

## 3.1 Approach

To evaluate the performance of different models consistently, a fixed subset of reviews was selected randomly from the dataset. Specifically, 50 reviews were randomly chosen as queries to test all models, balancing computational feasibility with robust evaluation.

This fixed sample size ensured that every model processed the same set of queries, thereby eliminating variability introduced by differences in the input data. The selection of 50 reviews was motivated by practical constraints: larger query sets would have significantly increased computational time and resource usage, particularly for transformer-based models like BERT and Sentence Transformers, which are computationally intensive.

For each query, the review and its associated hotel were removed from the dataset to prevent self-recommendation. The models were then tasked with identifying the most similar hotel based solely on the text of the reviews, and their performance was evaluated using the Mean Squared Error (MSE) between the predicted and actual ratings, as further detailed below.

## 3.2 Models

After implementing the baseline, we first tested the `BERT model` as direct competitor. But after assisting that its performances where, on average, no better than the baseline, sentence transformers were tested instead, giving better results, as further detailed below.

### 3.2.1 Baseline: BM25

BM25 is implemented as the baseline model using the `rank-bm25` library. It scores documents based on their term frequency (TF), inverse document frequency (IDF), and document length. For a given query, BM25 retrieves the most similar document and evaluates the recommendation based on the Mean Squared Error (MSE) between the predicted and actual ratings.

### 3.2.2 BERT

A pre-trained `bert-base-uncased` model is used to generate embeddings for reviews. The embeddings are computed by averaging the last hidden states of the model. The similarity between the query and reviews is calculated using cosine similarity. Precomputed embeddings are stored on disk for efficiency, avoiding redundant computations.

### 3.2.3 Sentence Transformers

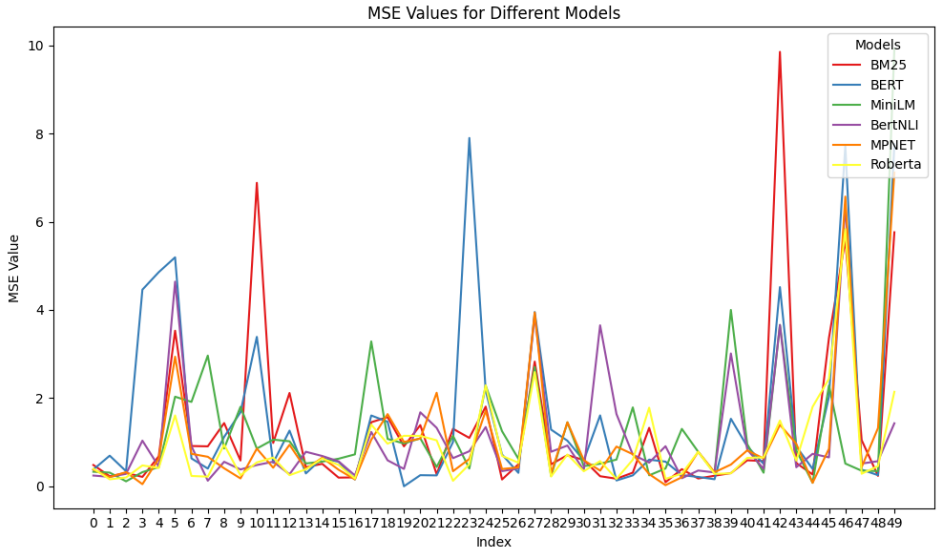
Several sentence-transformer models were evaluated, specifically:

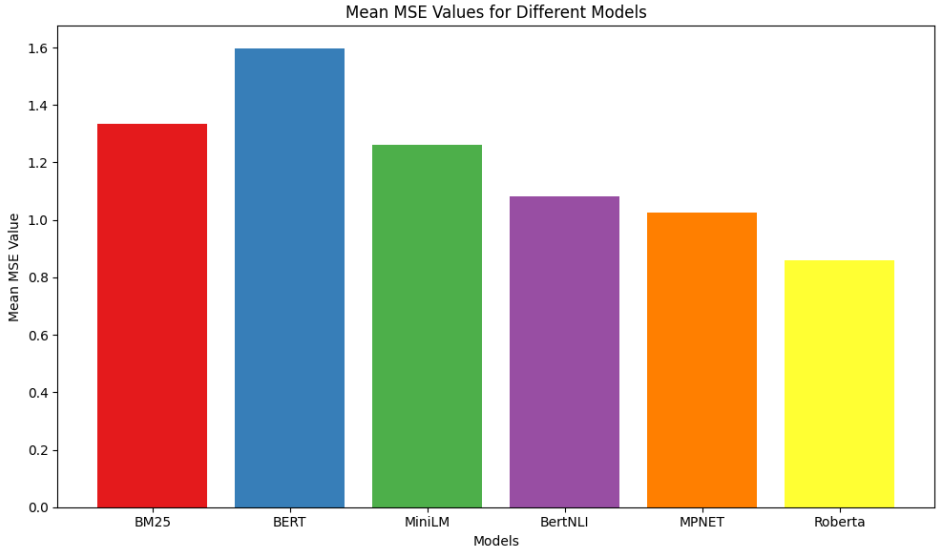
1. **MiniLM**: A lightweight model optimized for speed.
2. **BertNLI**: A variant fine-tuned on natural language inference tasks.
3. **MPNET**: A more advanced transformer with better contextual understanding.
4. **Roberta**: A large model fine-tuned for semantic similarity.

Each model precomputes embeddings for all reviews and uses cosine similarity for recommendation.

# 4 Results

As state above, sentence transformers models delivered best result, compared to BERT and the baseline. The best performing model was `Roberta`, with an average MSE of `0.86` score.
The following graphs show the MSE computed for each of the 50 queries, and the resulting average MSE for each model.



# 5 Ensamble Learning

At the end of this analysis, the final TripAdvisor recommendations are computed combining all results of all models with an Ensamble Learning algorithm.
For each query in input, the algorithm looks for the model that performed the lower MSE and chooses the corresponding recommendation as the best.

| Query Offering ID | Suggested Offering ID | Best Model | Best MSE |
|---|---|---|---|
| 99441 | 223713 | BertNLI | 0.24 |
| 93340 | 3432111 | Roberta | 0.16 |
| 93517 | 630950 | MiniLM | 0.11 |
| 111507 | 1516481 | MPNET | 0.05 |
| 115617 | 80771 | Roberta | 0.41 |
| 112328 | 99051 | Roberta | 1.61 |
| 1027494 | 99535 | Roberta | 0.23 |
| 1858565 | 73445 | BertNLI | 0.13 |
| 81212 | 1134288 | MPNET | 0.41 |
| 556822 | 100556 | MPNET | 0.18 |
| 240082 | 99514 | BertNLI | 0.48 |
| 1477545 | 111490 | MPNET | 0.42 |
| 81295 | 99518 | Roberta | 0.25 |
| 249712 | 113928 | BERT | 0.29 |
| 77931 | 262330 | BM25 | 0.51 |
| 93543 | 93507 | BM25 | 0.20 |
| 126257 | 112019 | BERT | 0.16 |
| 1174533 | 95263 | MPNET | 1.06 |
| 1235890 | 577590 | BertNLI | 0.59 |
| 282698 | 98678 | BERT | 0.00 |
| 1149434 | 93614 | BERT | 0.25 |
| 126402 | 217844 | BERT | 0.25 |
| 675616 | 571835 | Roberta | 0.12 |
| 102550 | 223724 | MiniLM | 0.40 |
| 1486164 | 112019 | BertNLI | 1.34 |
| 84079 | 99352 | BM25 | 0.15 |
| 93486 | 609602 | BERT | 0.30 |
| 1147514 | 2156247 | Roberta | 2.59 |
| 1091941 | 2253206 | Roberta | 0.22 |
| 217844 | 81192 | BM25 | 0.71 |
| 88186 | 102466 | Roberta | 0.34 |
| 1486164 | 1218792 | BM25 | 0.23 |
| 119721 | 2008152 | BERT | 0.13 |
| 109367 | 223755 | BERT | 0.25 |
| 112132 | 320058 | MiniLM | 0.25 |
| 124048 | 76061 | MPNET | 0.03 |
| 77638 | 1189576 | BertNLI | 0.18 |
| 87656 | 630409 | BM25 | 0.18 |
| 223830 | 81985 | BERT | 0.16 |
| 81241 | 1091941 | Roberta | 0.29 |
| 98655 | 262330 | BM25 | 0.59 |

*Continued on next page*

| Query Offering ID | Suggested Offering ID | Best Model | Best MSE |
|---|---|---|---|
| 115625 | 219703 | MiniLM | 0.31 |
| 113229 | 91490 | MPNET | 1.39 |
| 674743 | 235228 | BertNLI | 0.43 |
| 224948 | 3506933 | MiniLM | 0.08 |
| 98954 | 82976 | BertNLI | 0.66 |
| 93379 | 239357 | MiniLM | 0.52 |
| 84073 | 120072 | Roberta | 0.28 |
| 2516677 | 81192 | BM25 | 0.24 |
| 84127 | 122494 | BertNLI | 1.43 |

| Model name | #BestMSE |
|---|---|
| Roberta | 11 |
| BertNLI | 9 |
| BERT | 9 |
| BM25 | 8 |
| MPNET | 7 |
| MiniLM | 6 |

Table 2: Number of lowest MSE per model

# 6 Conclusion

This project demonstrates the effectiveness of leveraging advanced NLP techniques for recommendation systems. While BM25 provides a robust baseline, transformer-based models like MPNET and Roberta outperform it in terms of semantic understanding and accuracy.

Preprocessing is a critical step for improving model performance. Transformer models, despite being computationally intensive, provide substantial gains in accuracy. Efficient handling of embeddings (e.g., precomputing and storing) is essential for scaling these models to large datasets.

# 7 Delivery

As mentioned above, at each step of the execution, the relevant results (processed dataset, models embeddings, training results...) were stored to the disk, so that they could be retrieved at a later time without being recomputed. Adjustments were made so that these stored files can be downloaded automatically at runtime by anyone running the notebook, allowing the code to be easily run without the need to entirely recompute all the intermediate steps, which would require too much time.