NLP Project 1

TripAdvisor Recommendation Challenge - Beating BM25

Léo RINGEISSEN & Santiago MARTIN

# Objective

The primary objective is to develop a recommendation system based on TripAdvisor reviews for hotels. The goal is to be able to query the model by feeding it the compilation of one hotel’s reviews, and then getting a recommendation for the hotel most similar to the queried one. The completion of this objective contains several parts

First, we’ll build this recommendation system out of a baseline BM25 model, with no tuning and without performing traditional NLP preprocessing techniques on the training and input data. And second, we’ll test out different improvements to try to beat the initial baseline BM25 model, hence the challenge. This will be done by applying those previously mentioned preprocessing techniques and by testing other NLP models like TF-IDF, BERT, and T5 (small and baseline versions.

# Recommendation system functionality

## Training

We’re training the models on a corpus, which contains the compilations of all the reviews left by users for each hotel. So, each element of the list is one long string that contains every review stitched together for one hotel. This corpus is what **WILL** **NOT** be preprocessed using traditional NLP techniques initially for the baseline BM25 model and **WILL** be for attempting to beat the baseline BM25.

## Evaluation

To evaluate the models, we’ll use the ratings of the hotels. Each hotel in our final preprocessed dataset contains the average score for all 7 of the following rating categories: “service”, “cleanliness”, “overall”, “value”, “location”, “sleep quality”, “rooms”. To test the quality of the recommendations from a model, we perform several steps:

* Firstly, we query it 100 times with 100 hotels chosen at random (for each model we use the same random batch to ensure a fair comparison)
* Secondly, for each query we compute the Mean Squared Error of the difference between the 7 respective ratings of the queried hotel and the recommended hotel
* Finally, we compute the average the MSE over the 100 queries to obtain a final result to judge the quality of the model’s recommendations

# Data preparation

We initially disposed of an “offerings” dataset and a “reviews” dataset. “Offering” contained information for each hotel (name, location, ID, etc…), whereas “reviews” contained each review for all of the hotels, along with the ratings associated to that review and the ID of the hotel being reviewed.

The “offerings” dataset was not very useful since we’re merely interested in the reviews and ratings and the “reviews” dataset already contains a unique hotel identifier column. As a result, we only utilize “reviews”.

As mentioned previously, our final reviews table contained one line per hotel, each containing a compilation of the reviews for that hotel and an average aggregate of its ratings, to transform the initial reviews dataset, we had to perform several steps:

* Remove all unnecessary columns
* Extract ratings dictionary from ratings column based on our 7 categories of interest (“service”, “cleanliness”, “overall”, “value”, “location”, “sleep quality”, “rooms”) and turn them into columns for each review
* Summarize the table by hotel using the unique hotel identifier in each review
  + For this we compiled each review for one hotel as one long string
  + And we aggregated the ratings into averages of each of the 7 categories