Sentimental Hotel Review Retrieval System

Semester Project Report

Group Members:

Syed Zain Abbas Shah 231217

Muhammad Uzair Asif 231177

Muhammad Husnain 231147

Course: Information Retrieval

Semester: Spring 2025

Instructor: Ma’am Faiza Qamar

Institution: Air University Islamabad

Department of Creative Technologies

Contents

Abstract

This project presents an Information Retrieval system designed to enhance hotel search by focusing on user generated reviews rather than static hotel descriptions. Hotel data from multiple cities in Pakistan was collected using the Google Maps API, including details such as hotel names, addresses, ratings, google map links and user reviews.

The reviews were preprocessed and analyzed using Hugging Face Transformer models to perform sentiment classification. Sentence-BERT was used to generate semantic embeddings of the reviews, which were then indexed using FAISS for efficient vector based retrieval.

A Streamlit based interface allows users to input natural language queries and retrieve the most relevant reviews based on semantic similarity and sentiment.

1. Introduction

Problem Statement

Most hotel search systems rely on predefined hotel descriptions, ratings, or amenities provided by the hotels themselves. These descriptions often fail to capture the actual experiences of guests. As a result, users may make decisions based on incomplete or biased information that does not reflect real-world feedback.

Motivation

When users search for hotels online, they often want to know what others genuinely experienced. By enabling search based on user reviews, we aim to provide a more transparent and experience-driven hotel discovery tool.

Objective and Scope

The objective of this project is to design and implement a semantic and sentiment hotel review retrieval system. The system allows users to input natural language queries and retrieve relevant hotel reviews based on both the meaning of the query and the sentiment of the review content.

The scope includes:

- Scraping hotel data across multiple cities in Pakistan
- Preprocessing and sentiment analysis using NLP models
- Embedding and indexing reviews using Sentence-BERT and FAISS
- Building an interactive search interface with Streamlit
- Evaluating system performance using a manually curated ground truth

2. System Architecture

Tools & Technologies Used

- Python –programming language
- Hugging Face Transformers – For sentiment analysis using pre-trained models
- Sentence-Transformers – For generating semantic embeddings of reviews

- FAISS (Facebook AI Similarity Search) – For fast similarity-based search using vector indexing
- Streamlit – For building the web-based interactive search interface
- Google Maps API – For hotel data scraping across multiple Pakistani cities
- Google Custom Search API – To fetch and display relevant hotel images

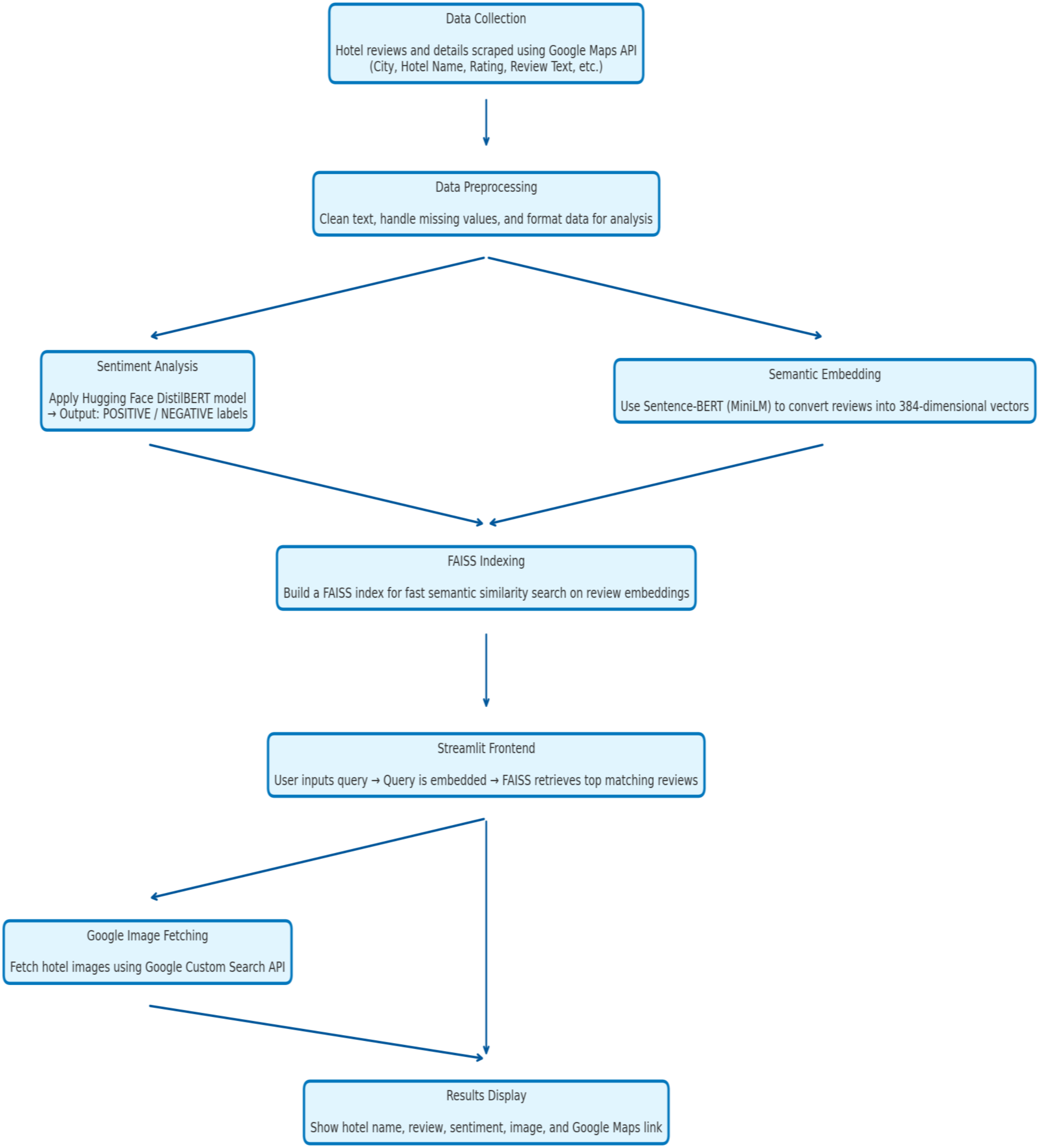
Architecture of the System

The system architecture is composed of four primary components:

1. **Data Collection Module**
   Scrapes hotel information and reviews from Google Maps for selected cities in Pakistan.
2. **Preprocessing and Sentiment Analysis Module**
   Cleans the review text and applies sentiment classification using the distilbert-base-uncased-finetuned-sst-2-english model. Then made a column of predicted sentiment in the dataset to store the predicted sentiments
3. **Semantic Embedding & Indexing Module**
   Converts cleaned reviews into dense vectors using the all-MiniLM-L6-v2 model and stores them in a FAISS index for semantic retrieval. Then stored the embedding in the embeddings column.
4. **Search Interface Module**
   Uses Streamlit to allow users to input natural language queries, performs semantic search using FAISS, and displays matching reviews with sentiments and hotel info.
5. **Image Fetching (Google Custom Search API)**
   For each retrieved hotel, a corresponding image is fetched using Google Custom Search to enrich the user interface with visual context.

Workflow Pipeline

The diagram below illustrates the full architecture of the hotel review retrieval system, including data collection, preprocessing, sentiment analysis, semantic embedding, FAISS indexing, user interaction through Streamlit, image fetching using Google Custom Search API, and final result display.



Data Collection

Hotel reviews and details scraped using Google Maps API
(City, Hotel Name, Rating, Review Text, etc.)

Data Preprocessing

Clean text, handle missing values, and format data for analysis

Sentiment Analysis

Apply Hugging Face DistilBERT model
→ Output: POSITIVE / NEGATIVE labels

Semantic Embedding

Use Sentence-BERT (MiniLM) to convert reviews into 384-dimensional vectors

FAISS Indexing

Build a FAISS index for fast semantic similarity search on review embeddings

Streamlit Frontend

User inputs query → Query is embedded → FAISS retrieves top matching reviews

Google Image Fetching

Fetch hotel images using Google Custom Search API

Results Display

Show hotel name, review, sentiment, image, and Google Maps link

3. Data Collection & Preprocessing

Data Source

We used the google maps API to collect hotel data from multiple cities across Pakistan. For each city, multiple keyword based queries (i.e, "hotels in Murree", "luxury hotels", "guest houses″) were executed to ensure a wide coverage of hotel types. For each hotel, we retrieved detailed information including the name, address, rating, phone number, total reviews, and individual user reviews.

Cleaning and Formatting

The raw data included multiple entries per hotel, each containing user's reviews. We removed duplicate rows, ensuring that reviews do not repeat, Then we handled null values

4. Sentiment Analysis

For sentiment classification, we used the distilbert-base-uncased-finetuned-sst-2-english model from Hugging Face Transformers as it is lightweight and version of BERT, fine-tuned for sentiment analysis tasks, making it efficient and accurate for real-world review data.

The cleaned hotel reviews were first checked for missing values and standardized into a consistent text format. Each review was then passed through the DistilBERT model using Hugging Face's pipeline("sentiment-analysis"). Only the first 512 characters of each review were considered to ensure compatibility with the model's input limits.

The model returned a sentiment label for each review. This label was stored in a new column, Predicted Sentiment, which was then used for filtering and displaying emotionally relevant results during search.

5. Semantic Search Using FAISS

Embedding Model

To know the meaning of user reviews and queries, we used the all-MiniLM-L6-v2 model from the Sentence Transformers library. This model converts sentences into fixed-size 384-dimensional vector embeddings that preserve semantic similarity. It is a lightweight and efficient.Each review in the dataset is transformed into an embedding during preprocessing

Indexing Method

We used FAISS (Facebook AI Similarity Search) to build a vector index from the review embeddings. The IndexFlatL2 method is used to perform the similarity search based on Euclidean L2 distance. The FAISS index is dynamically built at runtime using the review vectors and is capable of handling thousands of reviews with low latency.

Query Handling

When a user inputs a query:

1. The system first embeds the input query
2. The query vector is compared against all review embeddings in the FAISS index.
3. The system retrieves the most similar reviews based on distance and filters them based on the selected sentiment label.

4. The top-k most relevant reviews are returned and displayed to the user.

6. Evaluation

Ground Truth

For evaluation purposes, we manually labeled a subset of the dataset, hotel reviews from Islamabad with relevance labels corresponding to common search queries. Each review was labeled with up to five relevant tags label_1 to label_5 This served as the ground truth to measure the accuracy of our semantic search engine.

Metrics

We evaluated the performance of the retrieval system using the following standard Information Retrieval metrics:

- Precision: The fraction of retrieved reviews that are relevant .
- Recall: The fraction of all relevant reviews that were retrieved.
- F1 Score: The harmonic mean of precision and recall.

Formulas:

- Precision = Relevant Retrieved / Total Retrieved
- Recall = Relevant Retrieved / Total Relevant
- F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

Results

The system was tested on 10 queries. Each query returned the top 40 hotels' reviews. Following results:

Query	Precision	Recall	F1 Score
friendly staff	0.2250	0.0657	0.1017
clean rooms	0.5000	0.1408	0.2198
great location	0.1250	0.1163	0.1205
value for money	0.0250	0.0270	0.0260
poor management	0.0000	0.0000	0.0000

As the data was too much, and top-k was set to 30 so it retrieved fewer reviews, which caused bad accuracy.

Also, some data was not properly labeled as we got less time in the end.

7. Conclusion

This project shows how hotel search can be improved by letting users search through real guest reviews instead of static descriptions. By using sentiment analysis and semantic search, the system returns more relevant and meaningful results based on user intent.

While the model worked well for many positive queries, limited labeled data and some mismatches affected performance. Still, the project successfully demonstrates how NLP and vector search can make hotel search smarter and more user-focused.