Optimizing Restaurant Recommendations through Sentiment Analysis

by

Tasnia Juha
23241038
Shaira Chowdhury
20101261
Sultana Marium Trina
20101059
Md Fardin Rahman Ami
20101549
Abu Obaida Hasme
23341057

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
March 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

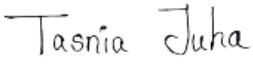
Student's Full Name & Signature:

Shaira Chowdhury
20101261

Sultana Marium Trina
20101059

Md Fardin Rahman Ami
20101549

Abu Obaida Hasme
23341057



Tasnia Juha
23241038

Approval

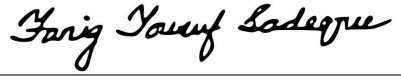
The thesis/project titled "Optimizing Restaurant Recommendations through Sentiment Analysis" submitted by

1. Shaira Chowdhury (20101261)
2. Sultana Marium Trina (20101059)
3. Md Fardin Rahman Ami (20101549)
4. Abu Obaida Hasme (23341057)
5. Tasnia Juha (23241038)

Of Spring, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on March, 2024.

Examining Committee:

Supervisor:
(Member)



DR. Farig Yousuf Sadeque
Assistant Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Thesis Coordinator)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

The rapid growth of the food industry and dining-out culture has led to a vast growth in the number of restaurants over the years. As a result, customers are overwhelmed with too many options to purchase their meals from. Using sentiment analysis, this study's goal would therefore be to create a data-driven recommendation system that can provide customers with the best restaurants that serve their preferred cuisine. Unlike previous research, this study includes a model that analyzes reviews in four languages namely English, Bangla, Code Switch (Bangla and English) and Code Mix (Banglish) from a multitude of food and applications. A customer's choice of restaurant will depend on multiple factors such as the recommendations of friends or food critics, their culinary preference, pricing, location, the general reputation of the restaurant and so on. This study proposes a novel approach that uses sentiment analysis based on primarily sourced restaurant reviews and ratings to provide personalized restaurant recommendations to interested customers. The assessment of our proposed model will be wide-ranging, through sentiment analysis applying multiple BERT model variants such as BERT for English annotated reviews, BanglaBERT for Bengali reviews, BanglishBERT for Code-Mixed reviews and XLM-RoBERTa for Code-Switched reviews with 79% , 74%, 75% and 77% best model accuracies respectively. Furthermore, this research investigates the analysis of various LSTM architectures, including Attention-LSTM, Bi-LSTM, and a Transformer-based T5 model. Utilizing customer reviews for a variety of restaurants, the efficiency of the model across multiple languages and types of sentiment analysis tasks will be evaluated throughout the entire set of experiments. The results of this study should provide a time efficient solution for food enthusiasts and the general consumer to be introduced to the finest restaurants with the best gastronomic delights, magnificent ambience and atmosphere and the best customer service.

Keywords: Restaurant Recommendation, Sentiment Analysis; Customer Rating; Customer Review; Cuisines; BERT, LSTM.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In the modern era, finding the ideal restaurant for quality food and spending quality time in a serene ambiance has become a matter of struggle. The growing number of dining options and the increasing importance of personalized experiences have led to a situation where conventional approaches, such as word-of-mouth recommendations and online reviews, can often be perplexing and misleading. There are several additional factors associated with this issue. For instance, a person's taste buds, preferences for foods, and surroundings may differ from those of their relatives and acquaintances. A friend or relative can recommend a restaurant simply because they enjoyed the food and service there. However, if one visits the restaurant based on this recommendation, they may not be satisfied. It can happen for a variety of reasons, such as the friend preferring a cozy place for dining while the next person loves a well-decorated restaurant where he/she can snap good quality pictures with an exotic background while also enjoying the meal. Determining a user's preferences and trends, and then creating a personalized recommendation based on this information, is required to give the user their expected experience. To make the recommendation more effective and efficient by using customers' preferences, interests, and analogies to ensure that they enjoy the desired experience, we are introducing this customized recommendation system from which consumers can receive an integrated recommendation.

Restaurant Business in Bangladesh:

The beginning of the 1990s saw an emergence of restaurant culture. This new type of food culture has changed Bangladesh over the years. In Bangladesh, the very first fast food restaurant opened its doors on Dhaka's Bailey Road .[9], resulting in a growth of fast food restaurants throughout the city of Dhaka and now in the whole country. At first, leaders in Bangladesh's fast food sector innovation were local business owners, and now international brands, such as Pizza Hut, Domino's, Burger King, KFC, etc. are entering the market too. With international fast food franchises also entering Bangladesh in the 2000s, it kept increasing. Based on the data of the BBS (Bangladesh Bureau of Statistics data) .[30], "In 2009-10, the number of hotels and restaurants was 2.75 lakh. With a 58.46% growth, the number is

now 4.36 lakh". One of the main reasons for this growth in this sector is that it has very little capital [30]. This makes the hotel-restaurant business very profitable and famous in Bangladesh. Therefore, this sector's economic contribution to the nation is likewise growing.

Importance of Reviews and Ratings in Restaurant Business:

In Bangladesh, customers are more likely to select their preferred restaurant over other restaurants based on the online reviews they see. As there are many restaurants, customers have many options too but mainly the reviews of these restaurants play a key factor for customers to choose which place they want to go to. There are many ways customers see reviews before choosing a restaurant such as Facebook Food Bank group, Online pages like TripAdvisor or equivalent, Google Reviews, etc. and they can also see ratings in places such as: Foodpanda, HungryNaki, or other relevant applications. Before going to any restaurant, the first thing customers do is check the reviews of that place. No matter how many items there are or how delicious foods they show, customers get the confidence to go there after looking at their reviews and ratings. A recent study .[13] stated that " Over 90% of customers before going to any restaurant read online reviews". If they receive good reviews then they get the confidence to try out their food. And in return, the restaurants amass positive reviews to win over customers' confidence.

Recommendation Model:

A recommendation system is an AI algorithm, which uses big data to predict ratings of products and recommend them back to the users. It is mostly associated with machine learning. In simple words, recommendation systems behave like an AI-generated customer assistant that recommends products based on their interest, past purchases, etc. As a result, these recommendations lead customers to make decisions about whether they want to purchase or not. The main three types of Recommendation systems are collaborative filtering, context filtering, and content filtering. The recommendation of a recommender model depends on the type of data that it is dealing with. In regards to our work, we will use user input to find reviews that contain target words as well as connected words based on word similarity thus making use of collaborative and context filtering.

How to use the recommendation model in restaurants:

Nowadays, we can see restaurants almost everywhere in Dhaka. There might even be 10 to 15 restaurants in just one building. However, the problem is that only a few restaurants provide good quality food. Now to solve the issue, Recommendation systems play an important role. The restaurant recommendation system's objective is to offer users recommendations based on their tastes, preferences, location, and other necessary factors. For providing these recommendations the recommendation model uses reviews from different places such as Online reviews, Google Maps reviews, Facebook food group reviews, etc. and in turn uses it as the primary

dataset. It then suggests to the target user particular restaurants according to their preference of cuisine, particular food item, and preferred location associated with restaurants that have positive user feedback.

1.2 Problem Statement

Given that restaurants aim to maximize profits and gain valuable customer insights, it is crucial for them to effectively showcase relevant food items to achieve that. However, restaurants that struggle in this aspect are most likely subject to failure and setbacks. Also, most platforms suggest to users the top-rated restaurants based on locations closer to them, raw star ratings, and sometimes the suggested restaurants are even sponsored. This discredits any form of past user reviews based on text for which, these platforms fail to provide target users with recommendations that were based on real user sentiments.

The basic building blocks of a recommendation model include the integration of sentiment analysis into it. Sentiment analysis is a computational approach used to find out the sentiment or underlying mood expressed in a piece of text. The ultimate objective is to extract subjective information from the given input and to classify it in terms of sentiments.

Most online food ordering websites that lack an effective recommendation system frequently face several problems including ineffective marketing, limited discovery, and uneven competition in the market. A recommender system can serve as a marketing tool for restaurants by suggesting their dishes or promotions to relevant customers. Without it, customers may spend hours searching for the restaurant that perfectly fits their preferences only to be left disappointed by the quality of their choices. Customers are solely reliant on their own knowledge and search capabilities when the recommendation system fails to make use of user reviews which limits their ability to discover new or lesser-known food options, leading to a narrower range of choices. Thus, most customers miss out on exploring diverse cuisines or trending dishes they are unaware of. Similarly, restaurants with established reputations or higher marketing budgets usually dominate search results or customer attention. This can create an uneven field, making it difficult for smaller or lesser-known restaurants to compete on equal ground. A recommender system provides valuable insights into previous customers'' choice of food and or similar cuisine. With it, customers can cut down time on finding the perfect restaurant for themselves and instead get recommendations that are tailored to do it for them. Now all they can do is just sit back and enjoy the best cuisines from the top recommended restaurants.

This brings us to our main concern: *The absence of raw text review sentiments, the limited discovery of smaller restaurants, and uneven or unfair competition in the market due to the absence of the utilization of customer reviews and ratings in current recommendation systems.*

Our study would therefore aim to address this matter by using sentiment analysis on customer reviews to construct a recommendation model by providing restaurants that best match the user's choices of food and cuisine. In this way, the user can

get first-hand information from real-life customer reviews, not just star ratings, and even discover newer restaurants that they initially had not heard of.

1.3 Research Objective

In our work, we plan to use sentiment analysis on restaurant reviews and ratings to build a personalized restaurant recommendation model. Analyzing based on cuisines, ratings provided to the restaurants, and reviews for their services and dishes.

1. Understand the factors of customer satisfaction based on star ratings and text reviews, afterward classifying and clustering these elements based on different parameters such as sentiments, food type, and location.
2. Analyze reviews of restaurants that provide the customer's preferred cuisine using NLP and sentiment analysis.
3. Recommend the best restaurants with the highest sentiment value to help customers discover new food.
4. Making a fair recommendation system that provides exposure to less-popular restaurants and the most popular ones.
5. Make customers satisfied with the targeted recommendations and discover new restaurants within their preferred cuisine and location.
6. Make comparisons between text-based sentiments and raw star ratings to best avoid biases in recommendations.

Chapter 2

Literature Review

A literature review, by definition, is a comprehensive evaluation of previously published research papers, journal articles, books, and other relevant scholarly works that contribute to a research topic. The mark of good research is the inclusion of a literature review to provide an overview of the existing knowledge about the research topic and highlight areas that require further study, justifying the research paper. In our case, we have gathered and read many published papers relevant to our topic of research from sources such as Google Scholars, IEEE publications, ResearchGate, etc. In the following paragraphs, we will briefly summarize some of these papers.

Another paper [15] proposes a recommendation model that uses an Amazon camera review dataset to build a model that combines user-preference and interests profiles, that is modeled without social media information instead using heuristics extracted from review writing aesthetics, and reviewer credibility analysis model where user reviews are weighted using credibility values and the reviewer is scored with feature sentiment based on the trust factor, network importance factor, and expertise factor. After combining all of these, a robust product ranking is achieved and then recommended to users. The model used in this paper is called Credibility, Interest, and Sentiment Enhanced Recommendation-CISER model. The authors of this paper used text analytics for the extraction of features by focusing on unigram product features and bigram features. Using spaCy to measure the context similarity. One of our research's clustering approaches was inspired by this paper's use of spaCy and cosine similarity. Nevertheless, this model generated an F1 score of 49.3%, a Precision score of 52.6%, and a Recall score of 46.4% on combined unigram and bigram features, which were compared with other models such as TF-IDF, RAKE, and TextRank, and it was shown that CISER had the best score among all of them. In another paper, [22], the authors aimed to provide a recommendation model that will only recommend specific movies to the user using user profile information and use sentiment analysis on the user's review of the movie for future recommendations. The research focused on the two aspects of using sentiment analysis on the reviews and recommending similar movies to the user using similarity scope from cosine similarity.

In [27], the authors, Pin Ni, Victor Chang, and Yuming Li, used both reviews and ratings. They combined both the reviews and ratings and created a more accurate model based on LSTM and Spark-ALS. To produce a better solution for sentiment

polarity classification, the authors of this paper have used text sentiment analysis with a deep learning algorithm. Also, they used TF-IDF to create word features. For the fine-grained sentiment analysis on customer reviews mainly on online shopping platforms, the author used the LSTM network. In the pre-processing, they used the tokenizer to remove the numbers, stop words (such as: full stops) , and symbols from the text using the NLTK library. Then, for learning the word vector of data, they used word2vec. Then it is used in classification as input..

In [19], the author attempted to create a context-aware recommender system. Which will extract the food preferences of customers from the website TripAdvisor and recommend restaurants following these preferences. In this paper, the dataset is pre-processed by noun extraction and then food-noun filtering using Wordnet to cluster these food nouns using Hierarchical Agglomerative clustering. They then proceeded to create a similarity matrix by using Wordnet to find the similarity between nouns. The method of clustering in this paper is somewhat similar to the clustering approach that our paper is attempting. Furthermore, this paper also extracted context information from the user comments using sentiment analysis and used the restaurant menus as well to recommend relevant restaurants. Their best-performing model had a precision of 92.8% while recommending restaurants.

The paper [31] attempted bias detection in text using transformer models such as BERT and RoBERTa variants. They used the RedditBias database and fit it into four types of pre-trained transformer models for text data embedding. Among them, the all-mpnetbase-v2 (full BERT) and all-MiniLML6-v2 model (mini BERT) are taken from Python's sentence transformer (SBERT) package in the Hugging-Face repository. And the RoBERTa models were the all-roberta-large-v1 model (all RoBERTa) and the xlm-roberta-base model (raw RoBERTa). This paper used t-SNE for two-dimensional visualization of this data and then used KNN classifiers to cluster the bias types. This research found that all-MiniLML6-v2 performed best for their intended bias classification. In our paper later on, we will discuss in depth how we also use all-MiniLML6-v2 to create embeddings for clustering purposes. Another use of Sentence BERT (SBERT) embeddings can be seen in the paper [26] where they combined SBERT and Factorization Machines (FM) to create a hybrid recommendation system. This research compared many SBERT embeddings, including all-MiniLM-L6-v2 and paraphrase-MiniLM-L3-v2. These models were used on two datasets containing information about movies and their reviews. The distance between the clusters of these embedding spaces (intra-cluster distance) was used to determine that the paraphrase-MiniLM-L3-v2 model gave the best outcome.

Another study, [21], attempts sentiment analysis on a dataset containing noisy Bangla comments and assigns these texts a strong negative, moderate negative, neutral, moderate positive, and strong positive polarity label. This is similar to our research in the sense that our dataset also comprises noisy English and Bangla raw customer reviews. In this paper, the lexical feature is extracted using TF-IDF weighted scores on n-grams of the words, and the semantic feature is extracted using FastText embedding and random embeddings. The models used for [21] are Bidirectional-Long Short Term Memory (Bi-LSTM) and multi-lingual BERT (mBERT). Bi-LSTM combined with FastText embeddings having a precision, re-

call, and F1 score of 52.24, 63.09, 57.15, and mBERT having 49.58, 56.43, and 52.79.

Another paper, published in 2021, uses a Bangla dataset to train two new models known as BanglaBERT and BanglishBERT [25]. These two models were fine-tuned and then compared with pre-trained models such as mBERT, XLM-R base, and large, IndicBERT, and sahajBERT. This paper also highlighted the performance of BanglaBERT on sentiment classification and natural language inference against XLM-R (large). Findings showed that BanglaBERT performance tended to be better when fewer samples were used. This paper's major contribution also includes introducing new Bangla NLI and QA datasets introducing the Bangla Language Understanding Benchmark (BLUB) and showing that, in the supervised setting, BanglaBERT outperforms mBERT and XLM-R (base) by 6.8 and 4.3 BLUB scores, while in zero-shot cross-lingual transfer, BanglishBERT outperforms them by 15.8 and 10.8, respectively.

A paper, [28], tested the effectiveness of using Google Translations API (Google Cloud) as a supportive tool for NLP practices such as sentiment analysis and named entity recognition. The paper attempted to use Google Cloud as a solution for the lack of feasible Sinhalese language-specific NLP tools. This paper is relevant to our research because we also incorporate the use of Google Cloud, which will be explained in the pre-processing section. This paper's observations showed that the accuracy level of sentiment analysis with python library NLTK on samples of Sinhalese translated to English was between 55%-70% and the NER processing of translated samples involved Stanford CoreNLP and NLTK for NER testing and gave an accuracy of about 100%, leaving some exceptions.

Another paper using the FastText model, [11], proposed word representation for 157 languages using datasets composed of a mixture of Wikipedia and Common Crawl. The pre-trained word embedding model here was used to identify different languages from the dataset for splitting the dataset based on language. Such use of FastText is also relevant to what we are attempting in our research. In [11] the performance of FastText was compared with existing models such as langid.py and it was found that FastText was faster and more accurate in comparison to other models. The accuracy score of Fasttext proved to yield an accuracy and time on the Wikipedia dataset of 93.0 and 1.3 respectively while langid.py had 91.3 and 9.4.

In this study [14], the authors conducted a comparison of various algorithms on a restaurant review dataset to assess their performance. The results indicated that both classifiers demonstrated remarkable accuracy. Furthermore, a best algorithm with the highest accuracy was extracted. Another study [1] utilized a framework for attribute detection in the context of online restaurant reviews. The approach involved designing a semi-supervised machine-learning method that incorporated both topic detection and sentiment classification.

One research suggested that sentiment analysis operates at different levels, ranging from an overall assessment of sentiment in a document to attribute-level evaluation [5]. It can be approached in two ways. In the knowledge-based approach, sentiment can be analyzed using language processing algorithms or Lexicon methods [8]. On

the other hand, supervised machine learning algorithms learn the polarity (positive, negative, or neutral) of reviews from a dataset initially classified by humans [7]. One Study defines two steps of sentiment analysis which involves detecting text segments, such as sentences and also determining the polarity and strength of sentiment per dimension [4]. This study [24] analyzed sentiments in the FDS domain by comparing simple and hybrid DL techniques, specifically LSTM, Bi-LSTM, and Bi-GRU-LSTM-CNN. The predictions were explained using SHapley Additive exPlanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME). Researchers have recently developed posthoc methods such as SHAP and LIME to explain decisions made by DL classifiers [16] [18].

The research [23] conducted by Osman et al. establishes the groundwork for improving recommender systems. Through applying the techniques of contextual and sentiment analysis, the study is designed to address the difficulties presented by domain sensitivity and improve the overall quality of recommendations. The authors acknowledge the limitations of previous sentiment analysis methods in a complex domain and propose an integrated approach that integrates innovative sentiment-based modeling with contextual factors. Sentiment analysis combined with recommender systems has provided an innovative solution, enabling users' raw textual feedback to be transformed into insightful refined preferences. This technique allows developers not only to better understand customer needs but also to address the issue of domain sensitivity from which sentiment analysis recommender systems often struggle. To get around the problem of domain sensitivity, Osman et al. proposed a model in their article that would make sentiment analysis impose consideration to the context. The suggested model demonstrates improved recommendation quality compared to the conventional collaborative filtering approach, which, while noteworthy, requires further investigation. Given the various forms of feedback and user attributes, incorporating sentiment analysis through context-related data might significantly benefit recommender systems. This could be beneficial for multiple problem domains. To address the fluctuation of ambiguous vocabulary across domains and deliver a more well-versed and appropriate subset of recommendations, The authors demonstrate significant advancement in collaborative filtering methodologies by addressing the challenges of data sparsity and error. This improvement leads to more informative and accurate recommendations.

The research article [33] by Singh et al. presents an innovative approach that aims to enhance food recommendation systems. This study leverages the method of neural collaborative filtering with sentiment analysis, presenting it as a novel means of recommending food items based on people's tastes. Specifically, by analyzing data on transactions from Zomato and other platforms, this approach discovers the popularity of local foods and empowers restaurant owners and industry stakeholders to make informed decisions based on essential patterns. The authors also employed deep learning techniques to enhance recommendations and analytics. They applied customer reviews and ratings to derive valuable insights, prioritizing sentiment over numeric values. Therefore, recommendations are determined by more than just current trends, but rather by an individual's taste preferences. The presented paper is a step forward in recommendation research as it demonstrates how neural networks and sentiment can be combined to produce more precise and customer-focused food

recommendation platforms.

A thorough methodology for constructing, filtering, and annotating a dataset tailored for sentiment analysis within the framework of Bengali-English code-mixed texts is presented in the study [12] by Mandal, Mahata, and Das. The research addresses a significant gap in relevant resources for code-mixture analysis, as this mode of communication is prevalent in culturally and linguistically diverse countries. The authors emphasize the adverse impact of code-mixing on traditional natural language processing tasks. Their contribution lies in the development of hybrid systems utilizing rule-based and supervised learning models to excel at such tasks as language identification and sentiment analysis with impressive accuracy. The dataset created explicitly has significant value due to its high inter-annotator agreement and diverse metrics. This provides a novel opportunity for sentiment analysis research in code-mixed instances. Due to the extensive creation and annotation of the corpus and the detailed exploration of the code-mixing and sentiment phenomena, this study may serve as a fundamental resource for future scholars who wish to conduct sentiment analysis in comparable contexts.

Chapter 3

Workplan

People mostly post their feedback and ratings on platforms in our target region such as FoodPanda, Google Review, TripAdvisor and AsiaFirms. To fetch data from such platforms, we have built a scraping tool using Python's selenium library for four different platforms. We collected more than 64,000 raw data combined from all sources and used our automated annotation tool to annotate the reviews based on language type.

After the exploratory data analysis, we removed irrelevant features and pre-processed the texts based on languages to avoid error. We further split the English, Bangla, Code Switch (CS), and Code Mix (CM) reviews from the entire dataset and used a label encoder to label the sentiments. It is to mention that in our work, Code-Switched language means a switch between Bangla and English language and Code-Mixed language means a mixture of Bangla and English words transcribed in English. The English and Bangla reviews were separately processed and trained on transformer models. As for the CS and CM reviews, two approaches were followed. The first approach included translating CS reviews to either completely Bangla or completely English based on a higher percentage of data, and transliterating the CM reviews, then merging them with the pure English and Bangla reviews. The second approach involved using CS and CM, as they are in transformer models, to find their sentiments.

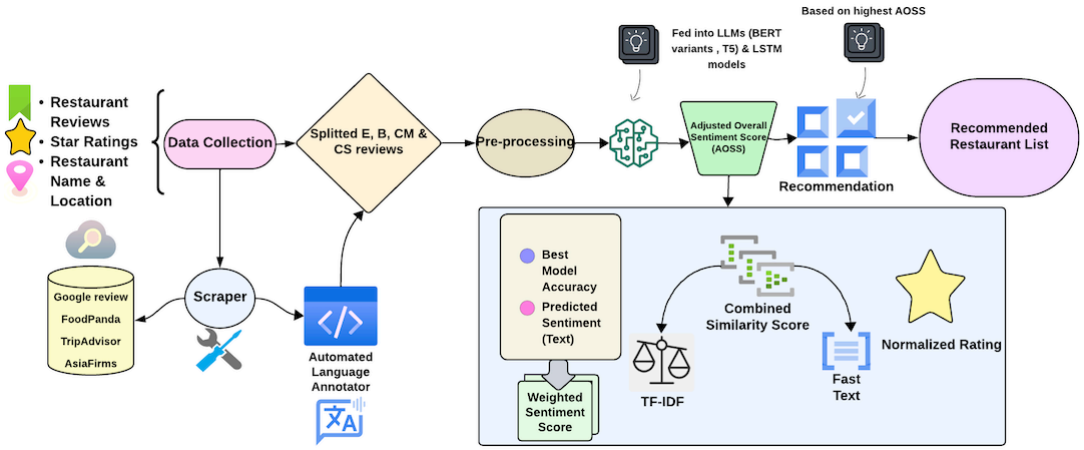


Figure 3.1: Workflow

We trained and evaluated large language models like BERT, Bangla BERT, T5, and Long Short-Term Memory (LSTM), a variant of RNN also used for comparison and analysis. We used these models to classify negative, neutral, and positive reviews of restaurants. This classification, along with raw star ratings, were then used to build a functioning recommendation system that clusters similar restaurants based on user preferences and recommends the highest rated restaurants.

Chapter 4

Description of the Dataset

4.1 Data Collection

We conducted our data collection procedure by means of the primary Data Collection approach, where we explored a decent number of restaurants inside Dhaka, which were taken from the following credible sources: Google Reviews, Foodpanda, Asiafirms, Facebook Page, HappyCow, Instagram and TripAdvisor. The initial data collection technique was administered manually, and we collected reviews from the previously mentioned sources. The process was hugely time consuming, prone to data entry errors and restricted scalability. Thus, our first immediate solution was switching to making use of the Chrome extension scraping tool "Instant Data Scraper." This made our data collection procedure much faster and convenient than before. The tool proved to be a much better alternative than the initial manual approach.

4.2 Scraping Methodology

According to our workflow, we planned to work with multiple text forms such as English, Bangla, English and Bangla mixed in text and words, so we could not find a suitable dataset that met our criteria. Besides, we focused on solving a real-world problem and decided to work with raw human data. We tried to collect secondary data, but the quality of the data we found was not as expected, so we rather decided to collect primary data on our own. From our research, we found multiple sources to collect data, such as browser extensions that help extract the reviews of users and various APIs; among them, a few were paid and a few were trail free access. From extensions, we tried "Instant data Scraper", "Data Scraper - Easy Web Scraping", "NoCoding Data Scraper - Easy Web Scraping" etc and from APIs, we explored "Google Maps Scraper", "Octoparse", "3i Data Scraping," and few others. By using these extensions and APIs, we faced a few major challenges. As a result, we did not get the exact data that we needed; it was giving the text data only, but the rating or stars were not given; a few APIs had usage and fetch limitations in a day, or in trail versions, the data we scraped was very limited in quantity. By using the extension, we had to scroll the pages and click to the next page and here also, it only gave us the data; after that, we had to rate the reviews manually. We found this approach very inefficient and time-consuming.

Platforms and accessibility

To avoid the challenges and drawbacks, we built our scraping tool to auto scroll and auto travel to the next pages and extract the reviews along with the ratings or stars, including the platform and restaurant names in the dataset. To scrape user review data, we chose "Google Map" reviews, "FoodPanda" reviews, "TripAdvisor" reviews and "AsiaFirms" reviews. These sites and platforms are widely recognized and primarily reliable in the Bangladesh region. We also tried to collect "Facebook" food page user reviews, but this platform had some strict terms and conditions for using their data, so we avoided this platform as they did not provide permission to use their data openly.

Tool configurations

We built four distinct scraping tools for four different platforms, each with different frontend codes and styles. Our tools are completely written in "Python 3" and use the libraries "Selenium" and "Pandas". Through "Selenium," we used drivers to control the browsers. We chose "Chrome Browser" as the browser and driver control. We experimented with code and setup with Chrome version "113.0.5672.24" onwards and found no issues. Our tool is compatible with the "13.0.5672.24" version and the latest ones (currently tested on the latest version "119.0.6045.159", Official Build). We also tested and tried to run our tools on different OS (operating systems) and found a smooth experience in MacOS ("Sonoma," "Monterey," "Ventura") and Windows (Windows 10, 11). Our code is written in IPNYB format, so we chose Jupyter Notebook" and "VS Code" for the IDE.

Challenges

While scraping data, we found challenges and obstacles afterwards. Therefore, we took a few steps to resolve them. These are:

1. Due to security reasons, robust platforms like "Google", often change their HTML and CSS codes inside the blocks of reviews. Changes are like modifying class names, tags, div (division) names, id names etc. So according to the changes, we also updated our code's "XPATH", "Class Name" and "CSS element" values.

2. Few sites have bot testing or, CAPTCHA issues. So we kept the sites logged out and collected the links in guest or, anonymous mode.

3. As we used this tool on distinct devices, we received few warnings because of different Python versions as few build-in functions were not available in old Python version. These issues got solved afterwards, and now our tool works on Python 3.8 and onwards.

This scraping tool made our data scraping faster and more efficient. Even though we faced several challenges, the solutions were achieved through extensive testing and refining, emphasizing the importance of continuous improvement in web data scraper development. Afterwards, we could easily scrape reviews, restaurant names and ratings just by providing the restaurant's site link. Moreover, we could easily regulate fetching data among them just by controlling the scrolling loop. So, we could adjust the range according to our needs.

4.3 Language Annotation

Upon constructing our scrapers, we achieved the capability to effortlessly extract textual reviews and Star Ratings as numerical values, ranging from 0 to 5. The names of the restaurants and the outlets were provided to our scrapers as input. However, complications became apparent when we were required to annotate the textual reviews. Because the reviews included a mixture of Bangla and English text, some texts contained a combination of both languages. Additionally, there were cases where the text was written in inappropriate English, as its intended meaning was in Bangla but transcribed into English. We observed the issue after we went over 2,000 text reviews and found that a significant number of the text reviews had been incorrectly classified. For example, some texts were quite large enough and written in English but switched to one or two Bangla words in between. As a result, it went unnoticed by us in the first annotation. To address this issue, we acknowledged the necessity of implementing an automated language detection system for the text reviews that we can incorporate into our scraper.

Incorporating ”nltk”, ”langid”, ”re”, ”pyspellchecker” and ”emoji” libraries, we created our automated language annotation tool. In our language annotation tool, punctuation and emojis were manually removed. We used the probability of misspelled words in a text, and if the percentage of correctly spelled words was greater than 85%, we annotated it by associating it with the regex pattern accordingly. Before passing it through the regex pattern we made sure to classify the text using ”nltk” and ”langid” libraries. A nested if condition helped us to annotate the ”E”(English), ”B”(Bangla), and ”CS”(Code-Switch) correctly. However, if the text was not classified as either ”E”, ”B” or ”CS”, we annotated it as ”CM”.

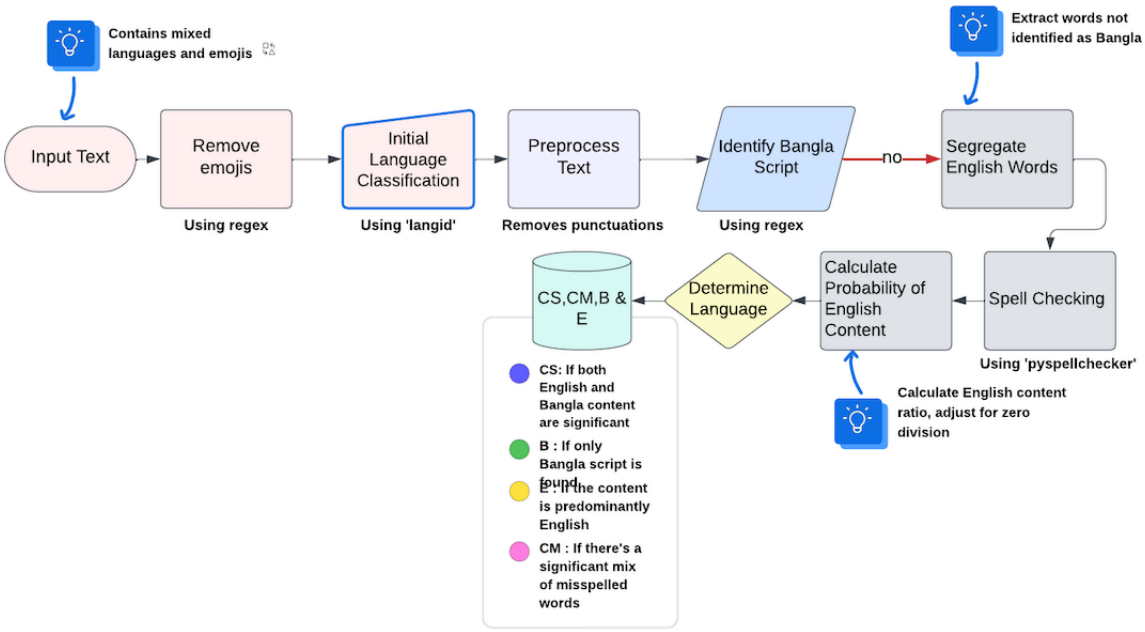


Figure 4.1: Workflow of Language Annotation

Initially, annotating the English texts seemed much easier. But as we explored more libraries and passed our dataframe through it, we observed that many English words were missing across different libraries. Also, as our data frame consisted of the reviews from real users across different platforms, many reviews consisted of misspelled words for which it was affecting our annotation. This issue was handled quite efficiently in our language annotation tool. For annotating "CM", it took us a lot of time to find a suitable approach as it was the most difficult part of the annotation. There was no specific libraries that handles "Banglish" words. That is why we needed to manually handle this case. However, there still remains some discrepancy in our tool for accurately detecting the "CM" reviews.

4.4 Overview of Dataset

In order to properly recognize and closely study consumer behaviors and sentiment with respect to certain restaurant types and other key features, we constructed our dataset by means of the primary data collection approach purposely collecting raw data from various credible sources. This aided us in analyzing qualitative data in correspondence with the quantitative ones consequently yielding outstanding visual representations of the results, more commonly known as the exploratory data analysis (EDA). Although the raw dataset initially carried a handful of inconsistencies, further cleaning, polishing and pre-processing the data eliminated those incongruities.

The dataset initially carried information related to restaurants, encompassing a total of 64,693 entries organized across seven distinct columns: Restaurant Name, Restaurant Type, Outlet (Dhaka), Review, Rating, Language and Source of Review. From among these seven columns, six columns hold qualitative or textual data while one column, 'Rating', contains numerical data. Before data cleaning, our dataset contained reviews of 371 unique restaurants inside Dhaka city, providing a diverse set of establishments for analysis and exploration. We grouped the 'Language' column's values into 5 language annotations for better classification of data for future work: E (English), B (Bangla), Code Mix or CM (Banglish), Code Switch or CS (Bangla+English), BE (we later converted it to CM and CS by tag annotation). The affirmation of the credibility of this dataset can be confirmed by acknowledging the collection of data from sources such as Google, Foodpanda, Asiafirms and equivalent. An integral feature of this dataset is the inclusion of ratings spanning from 0 to 5, providing a nuanced insight into the diverse range of experiences documented within the dataset. A tabular representation of the distribution of ratings corresponding to the number of reviews, Restaurant and Review types is as follows:

.

Ratings	Total Reviews
5.0	22,570
4.0	14,708
3.0	8,574
2.0	4,729
1.0	13,159

Table 4.1: Ratings and Reviews

Language	Total Reviews
E	51,868
B	6,112
CM	4,645
CS	1,115

Table 4.2: Language and Reviews

Sentiment analysis has been conducted on reviews in different languages, revealing insights into positive, negative and neutral sentiments across various language categories. Moreover, the 'BE' annotation was later classified into CS and CM. Therefore taking these current annotations and grouping into sentiments yielded us with the following results:

Language	Positive	Negative	Neutral
E	33,268	11,300	7,099
B	1,930	3,428	706
CM	1,440	2,559	616
CS	433	524	124

Table 4.3: Sentiment of Language Annotations

4.5 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a process that aims to simplify the understanding of data presented in columns and rows through visualization and interpretation. We also ensured maximum clarity by utilizing various visualizations. Our investigation focused on the distribution of reviews and ratings and other various aspects in our dataset that will be explored in this section of this research. The representation of positive, negative and neutral sentiments is as follows:

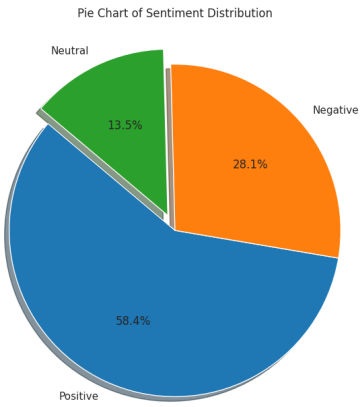


Figure 4.2: Sentiment Distribution

In Figure 4.2 the percentage of positive reviews can be seen to span a greater value of 58.4% with contrast to the remaining 2 sentiments. Interestingly, when we group separate representations to see the sentiment distribution per language, the English language reviews can be seen to have had a huge impact in the overall sentiment score being high in Figure 4.3:

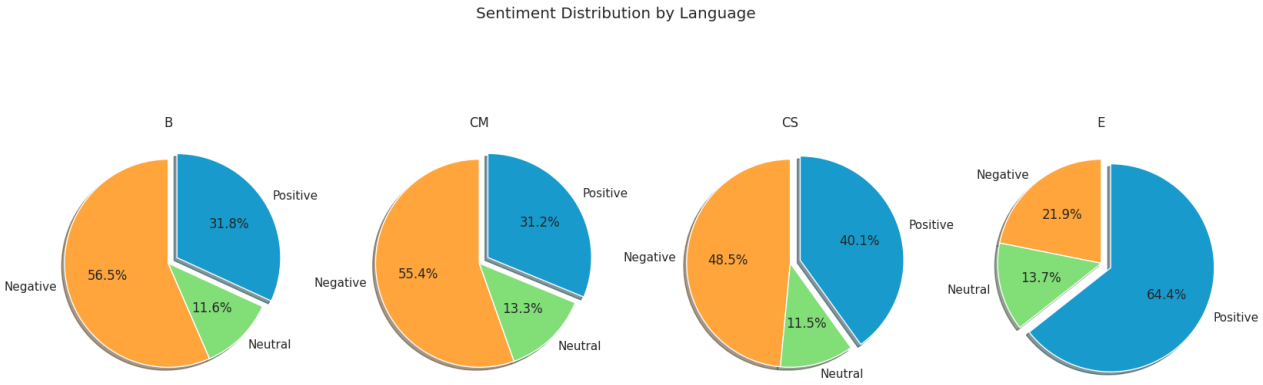


Figure 4.3: Sentiment Distribution per language

The star rating distribution values, represented in the bar chart of Figure 4.4, show the allocation of review counts corresponding to the star rating. While the 5 star rating count is the highest, 2 star rating count seems to be the lowest.

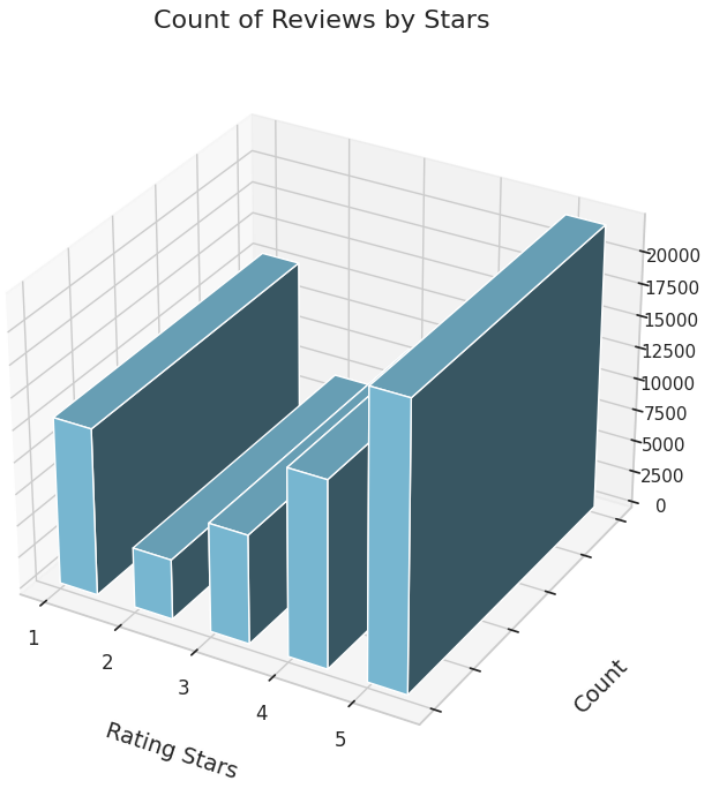


Figure 4.4: Star Rating Review Count

When we take into consideration our sources, the distribution of 5 star ratings in general are the highest except in Foodpanda.Looking at Figure 4.5, we can see that Foodpanda seems to have the highest percentage of 1 star ratings from among the rest of the sources. Similarly, while all other sources have a similar 1 and 2 star rating percentage, Foodpanda sees a huge difference between those two.

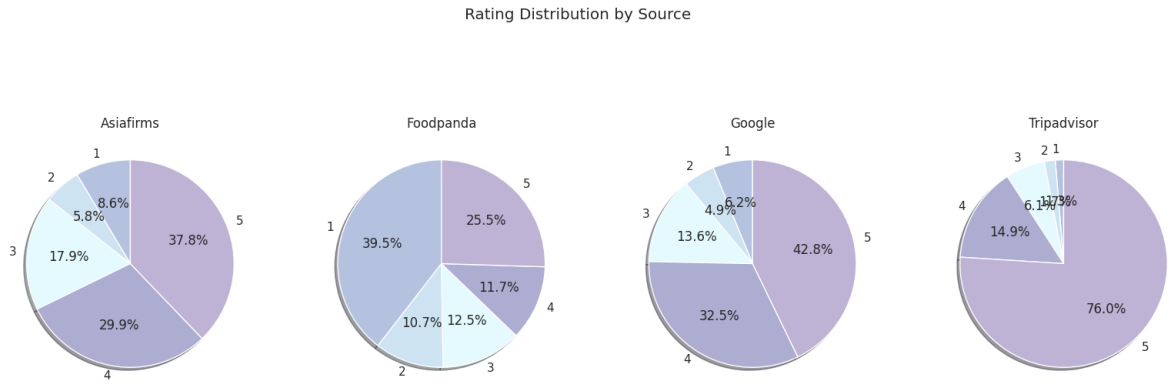


Figure 4.5: Rating Distribution by Source

On average, the distribution of rating counts as well as the count of average rating per outlet can be depicted in Figure **??**. The distribution of ratings show a fair allocation of rating counts. It is important to note that there are no extreme lows or highs of review counts per rating and ensures data credibility. Similarly, each outlet has a decent average rating value with all outlets having an average rating above 2.5.

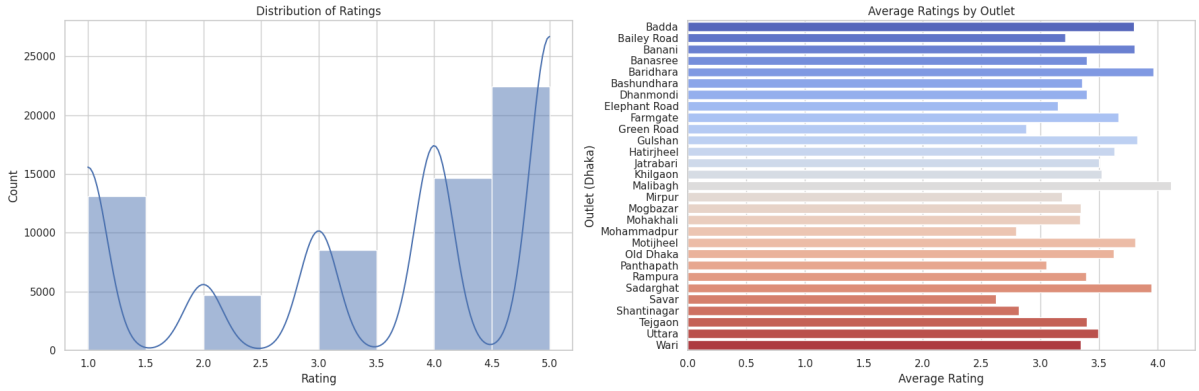


Figure 4.6: Rating Distribution

From among the total number of restaurants explored, the outlets were plotted against the number of unique restaurant count to get a clear picture of the scenario. It can be seen that Dhanmondi has the highest number of unique restaurants present while Malibagh, Sadarghat and Hatirjheel contain a very poor amount. The outlets with a higher number of unique restaurant counts are mostly situated at the heart of Dhaka, therefore the portrayal of the depiction is appropriate. Even though this is not the finding after exploring all the possible restaurants present in the target outlet such as Malibagh, but a representation of our own collected data.
The number of reviews, restaurant types and the average rating are some of the main key features of our dataset which hold the ability to draw a fair comparison

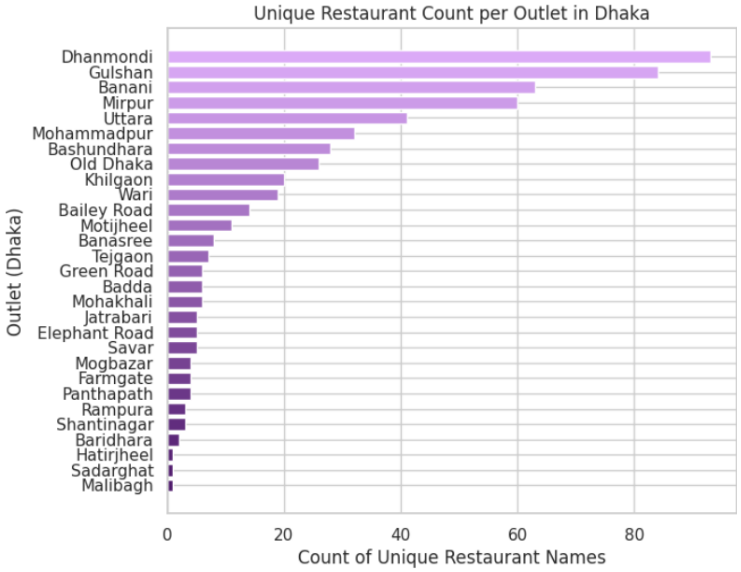


Figure 4.7: Restaurant Count per Outlet

analysis between one another. Figure 4.7 shows the visuals of unique restaurant count per outlet in Dhaka where we can see the dominance of Dhanmondi having the most outlets per restaurant.

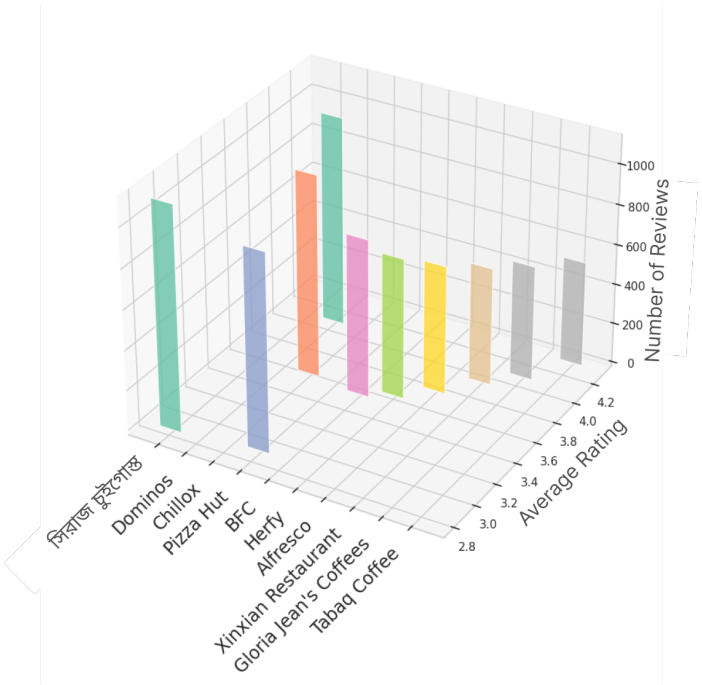


Figure 4.8: Top 10 Restaurants and Review Counts (3D Representation)

A 3D representation in Figure 4.8 illustrates the top 10 restaurants that have the highest review counts along with their average ratings. Here, the restaurant 'সিরাজ চুইগোস্ত' has more than 1000 reviews and 'Dominos' has more than 800 reviews.
From Figure 4.9, distribution of different types of restaurants are illustrated. The x-axis classifies restaurant types, from Bangladeshi to Shawarma Restaurant, and the count of each restaurant type is classified by the y-axis. The bars sizes reflect

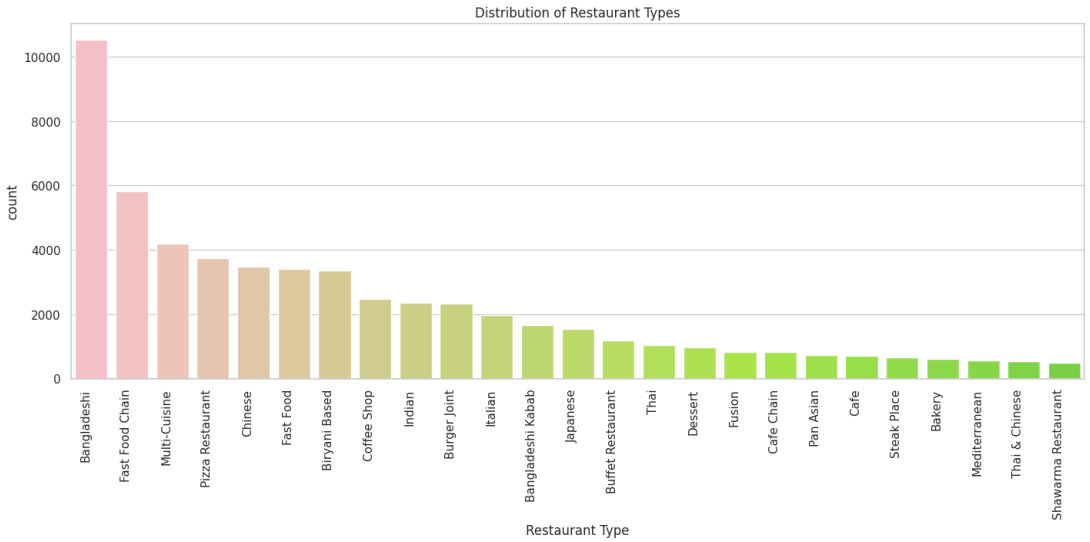


Figure 4.9: Distribution of Restaurant Types

the count of each restaurant type in the dataset.This is reflected from the graph that Bangladeshi restaurants are the highest in number when comparatively compared to others and also that Shawarma restaurant has the least. The longer bars compared to the categories such as Thai-Chinese, Mediterranean, or Steak Place, indicative of a low count, demonstrate the presence of fast food chains, multi-cuisine, and pizza restaurants.

Our cleaned dataset was used in the creation of Bengali, English, CM, and CS word clouds which provided us with a visualization of the most favorable words based on the sentiments of the corpus accordingly. For English. The results shown in Figure 4.10 are the representations after removing stopwords to avoid anomalies and highlight the importance of the most occurring word giving a general idea of the dataset. The words containing most importance can be reflected upon the clouds with bigger fonts. The words 'food', 'good', 'quality', 'taste', 'burger', 'pizza', 'restaurant' etc. are visible more prominently because these were used in the review text more often.

Figure 4.10: Word Cloud of Reviews in English

Figure 4.11: Word Cloud of Reviews in Bangla

Similiarly, Figure 4.11 illustrates that **খাবার** ,**ভালো** ,**মাংস** etc. words carries the most prevalence in Bangla dataset's review text. Interestingly, in Figure 4.13, we can see similiar occurences of the words **খাবার** ,**ভালো** like the Bangla dataset's wordcloud

along with the word চিকেন. In the same manner, Figure 4.12, depicts the wordcloud from CM dataset where 'chicken','khabar','valo' etc are the predominant words.

Figure 4.12: Word Cloud of Reviews in Code Mix (CM)

Figure 4.13: Word Cloud of Reviews in Code Switch (CS)

From the depictions of the wordclouds, it can be seen that words carrying sentiments have been massively illustrated, shining light on their importance and a rough portrayal of the dataset as a whole. Four separate word clouds return a separate output with the most occurring word being the largest. This aids us in understanding the importance of the words based on sentiments in the context of reviews and customer feedback.

Chapter 5

Data Pre-Processing

Pre-processing is an integral part of research which is required to prepare the data well for passing into specific models. This crucial step is needed to minimize errors, increase efficiency, and present an ideal visual to understand and analyze the results better.

The step of data cleaning was thoroughly necessary to achieve consistency in our dataset and make it standardized. Our dataset consists of 7 columns, 6 of which are qualitative while the remaining is quantitative. Since we collected reviews of restaurants based in Dhaka, our dataset holds reviews in English, Bengali, and even a mixture of both languages which were later annotated into further criteria for better categorization. Interestingly, some reviews collected via Foodpanda even consisted of Chinese dialects which were later dropped and considered anomalies.

The establishment of positive, negative, and neutral sentiments was based on grouping star ratings into positive, negative, and neutral categories where 4 and 5-star ratings were considered positive, 1 and 2 were considered negative, and 3-star ratings were considered neutral. However, we detected some inconsistent reviews such as 6, 2.5, and 0 star ratings which appeared due to the initial manual scraping and were later replaced with 5, 3, and 1 respectively. Afterward, we removed all the duplicate rows to get a clean and consistent dataset. The raw dataset originally consisted of several discrepancies one of which was the inclusion of the data, 'other' and 'BE' in the 'Language' column which were corrected using the language annotator. One-word reviews or reviews consisting of less than 4 words were removed to preserve data stability. Furthermore, due to manually typing restaurant types, names, outlets, and sources, the same values were interpreted differently raising incongruity. The standardization of restaurant types, names, outlets, and sources was thus carried out to fix these issues. The current dataset dimension now stood at 64,693 x 7. The current pre-processed dataset now consists of a total of 57,751 reviews with a representation of positive, negative, and neutral sentiments. Afterwards, we divided the pre-processing into two phases to feed into models. One for Bangla text and another for English text.

For the English text, our study explores the experiment that was conducted through a multi-step procedure to pre-process and construct some models for sentiment analysis on textual data. The dataset initially consisted of a column named 'Review'

that contained textual review, accompanied by appropriate labels denoting the sentiment. To facilitate the method of machine learning, we initially used a Label Encoder from the scikit-learn library to transform sentiment labels into numerical values. Afterwards, we narrowed our emphasis to reviews written in English by filtering the dataset appropriately. The text-cleaning procedure encompassed multiple stages to optimize the quality of the textual information. The text was normalized using a series of operations, including converting it to lowercase, removing non-alphabetic letters, eliminating stop words, and lemmatizing words using the NLTK toolkit. The objective of this method was to establish a uniform format for the text and decrease its complexity to maximize its suitability for analysis. After performing text preprocessing, we divided the dataset into training and testing while ensuring an equal distribution of sentiment labels in both sets. The training data was afterward tokenized using the BERT's tokenizer.

Before Pre-processing	**After Pre-processing**
After having a great experience in their chefs table branch, visited this food shop. To my utter surprise, they served cold BAO with cold fries. It was a big disappointment. 👎👎👎👎👎 Beef noodle was okay with flavors. 👍 Honey glazed chicken was so-so but has excessive sauce 👎👎	great experience chef table branch visited food shop utter surprise served cold bao cold fry big disappointment beef noodle okay flavor honey glazed chicken excessive sauce

Table 5.1: English Text Pre-Processing

We had to do the pre-processing separately for the Bangla reviews because those texts contained different alphabets and structures. Based on the language tag "B" (which denotes Bangla text), the Bangla texts were split from the entire dataset. We got a total of 6221 data from the Bangla dataset. As mentioned above we assigned the polarity labels to star ratings 1 and 2 as Negative, 4 and 5 as positive, and 3 as neutral. Then we encoded the sentiment column using "LabelEncoder" module from the scikit-learn library and found that the sentiments were encoded to 0, 1, and 2 as negative, neutral, and positive respectively. We used unicodedata, regex and string modules to clean the Bangla texts further. Since the Bangla reviews might contain symbols, punctuation, emojis, HTML tags, numerical characters (like '+','−'), specific Unicode Characters and Carriage Return and Newline Characters. Thus, to normalize the Bangla reviews we created a custom 'BanglaCleanText' function and systematically removed each of these with placeholder strings. Afterward, we removed all the stopwords from our normalized text using a custom set of stopwords as there were very few pre-existing Bangla stopwords available in BNLP's BengaliCorpus. The normalized and clean reviews are then split into train and text data in a ratio of 80:20 and stratifying based on the sentiment distribution. Then these reviews were tokenized using the AutoTokenizer class from the Hugging Face transformers library. We padded and truncated these texts to ensure consistency in all the Bangla reviews. Then we split the processed data to a ratio of 80:20 to prepare for training and testing.

.

Before Pre-processing	After Pre-processing
ফুল গ্রিল আর চারটা নান রুটির সাথে মেয়োনিজ আর সালাদ একটু বেশি করে দিতে বলছিলাম তারা মোটেও দেয়নি, এইসব ভন্ডামি ঠিক না,,, পার্সেল নিছি বলে কি এরকম | ধোঁকাবাজি করবেন,,, এর থেকে মোল্লা হোটেলেই ভালো😡😡😡😠	ফুল গ্রিল চারটা নান রুটির মেয়োনিজ সালাদ একটু বলছিলাম মোটেও দেয়নি এইসব ভন্ডামি না পার্সেল নিছি ধোঁকাবাজি মোল্লা হোটেলেই ভালো

Table 5.2: Bangla Text Pre-Processing

We also have CS (Bangla + English) and CM (Baglish) data to pre-process and clean. We used two approaches for these two language types.

Our first approach involved translation and transliteration using Google Translator API. Since the CS data had both Bangla and English words thus we used the language annotator to separate the CS data into Bangla and English based on the condition that if more than 50% of words in a review are Bangla then it would be labeled B otherwise it would be E. Then we used the Translator module from googletrans library to translate English words to Bangla for B-labeled reviews and to translate Bangla words to English for E-labeled reviews. As for the CM reviews, as they were Banglish words thus we found that using Google Translator API we could transliterate these words without translating them completely. Thus, after translation and transliteration, CS-B (CS translated to complete Bangla) and transliterated CM were passed through the same pre-processing process as the pure Bangla reviews and CS-E (CS translated to complete English) went through English pre-processing. In the second approach, the focus is made on the complex linguistic

Language	Type	Review
CS English	Raw	Burger cheese slice হাফ দিয়েছন কেনো? burger bun ফ্রেশ ছিলো না, শক্ত ছিলো। worst chicken cheese burger I had ever have👎👎👎
	Translated	Why was the Burger Cheese Slice Half? Burger Bun was not fresh, it was hard. Worst chicken cheese burger i had ever had👎👎👎
CS Bangla	Raw	প্রথমবার অর্ডার করেছি প্রাইস অনুসারে বার্গার খুবই মজা ছিল অবশ্যই তাদের অন্যান্য আইটেমগুলো ট্রাই করবো। Best wishes👌
	Translated	প্রথমবার অর্ডার করেছি প্রাইস অনুসারে বার্গার খুবই মজা ছিল অবশ্যই তাদের অন্যান্য আইটেমগুলো ট্রাই করবো সেরা শুভেচ্ছা👌
CM Transliterated Bangla	Raw	naga drums and pankha wing tah onek yummy specially Naga drums😛😊
	Transliterated	নাগা ড্রামস এবং পানখা উইং তাহ ওয়ানক মুখরোচক বিশেষভাবে নাগা ড্রামস😛😊

Table 5.3: Google Translator Outputs

structure of Bangla written in English transcript. The main purpose was to cat-

egorize CM reviews generated by the users into three sentiment classes: negative, neutral and positive, that is, based on their ratings assigned. We utilized a robust pipeline consisting of data pre-processing, sentiment analysis, and model evaluation, based on recent natural language processing (NLP) methods as well as transformer-based models. Our research began with the analysis of an existing GitHub repository – 'banglishbert' using a pre-trained language model that can capture the subtleties of Bangla syntax and semantics, especially when in an English context. This corpus is the basis of our research, in that it simply offers the language structure required to cope with Code-Mixed (CM) data. To categorize the sentiments, we mapped the ratings to sentiment labels like before. After that, the dataset was partitioned into training (80%) and validation (20%) sets allowing a full-scale evaluation.

Before Pre-processing	**After Pre-processing**
size boro Hoya uchit chilo 😕.Ei price range e choto Hoye gese😐	size boro hoya uchit chilo ei price range choto hoye gese

Table 5.4: CM Text Pre-Processing

The pre-processing step employed AutoTokenizer from the transformers library, pre-trained on "csebuetnlp/banglishbert". This tokenizer copes well with CM data and gives interesting tokenization information. For optimal performance of the model and to preserve the majority of the data, we truncate sequences at 69 tokens, which represents the 95th percentile length of sequences.

For the CS dataset, we did 2 types of processing, one for lexicon prediction and another for model prediction. Firstly for the lexicon approach, we removed non-alphabetic characters and all punctuation from the text ensuring only words were left. After that we converted all English words to lowercase, except the Bangla, to standardize the text and remove stop words in English and Bangla. Lastly, we split the text into tokens. For the model prediction approach[29], we pre-processed everything the same way except removing the stop words, and the text was not split into tokens. We also implemented a hybrid-based sentiment model where both of these pre-processing techniques were used. We used two pre-processing together and found results for each case.

Before Pre-processing	**After Pre-processing**
খাবার ভালো ছিল, But not the environment of the Place 😅.	খাবার ভালো ছিল but not the environment of the place

Table 5.5: CS Text Pre-Processing

Chapter 6

Models, Analysis and Clustering

6.1 Models

For this study, we opted the use of deep learning and implemented it using neural network models like Long Short-Term memory (LSTM) , Attention-based Long Short-Term Memory (ATTENTION-LSTM), and Bidirectional long-short term memory (Bi-LSTM) as well as natural language processing models such as Bidirectional Encoder Representations from Transformers (BERT) where we used BERT and T5 for the English text, BanglishBERT for CM dataset, mBERT and XLM-RoBERTa for CS dataset and BanglaBERT for the Bangla text to perform sentiment analysis on our primarily sourced dataset.

6.1.1 Bidirectional Encoder Representations from Transformers(BERT):

Bert's models help to understand ambiguous language by considering surrounding contexts. It leverages transformer-based neural networks and functions with bidirectional encode-decode representations, which were found to be effective in the case of human languages.

BanglaBERT:

We used the model BanglaBERT on our Bangla dataset as we found that the BangLUE score was higher in the Bangla Bert model than in other Bangla models. Thus, we used the benchmark model for the Bangla reviews. The setup of the model 'csebuetnlp/banglabert' [25] involved using the AutoModelForSequenceClassification class from the Hugging Face transformers library. The BanglaBERT model follows standard model architecture, with 12 transformer block layers (L-12) with each layer having a 768 units hidden size (H-768). And each transformer has 12 attention heads (A-12). This model has a total of 110619651 parameters and all of them were trainable. As mentioned in the pre-processing section, most of the normalization was done manually using modules, such as unicode data and regex, and custom-made stopwords set. Then the data was split in 80:20 ratio and tokenized and padded using AutoTokenizer to fit in the model. We trained the model with 25 epochs with a 0.000001 learning rate, with a dropout rate of 0.1 and the batch size was 16. The learning rate was determined after fine-tuning the model and

we introduced early stopping with a patience of 4 epochs to prevent overfitting of the model. The model was configured to a three-class classification thus the model classifies the raw texts into either positive, neutral, or negative text. The model was trained on the normalized Review, thus with no emoji, and Sentiment column, made from rating column. But we also generated a classification prediction from this model using only the unprocessed, raw Reviews.

BanglishBERT:

In this part, we present the training procedure and results of our sentiment classification model using the BERT-based architecture, specifically "BangishBERT" on raw CM datas. The dataset was preprocessed to convert ratings into sentiment labels, where ratings of 1 or 2 were labeled as 'negative', 3 as 'neutral', and ratings above 3 as 'positive'. This yielded a training dataset of 4,764 samples and a validation dataset of 1,192 samples. Here, we employed AutoModelForSequenceClassification class from the transformers library for sentiment classification utilizing the "csebuetnlp/banglishbert" model [25], with three output labels associated with the sentiment categories mentioned before.

Tokenization was performed with a 95th percentile sequence length of 69.25, which informed our choice to truncate or pad sequences to a maximum length of 70 tokens. The training was executed over 10 epochs with a batch size of 32, employing strategies such as logging per epoch, saving the best model based on accuracy, and a warmup period of 500 steps.

mBERT and XLMRoBERTa:

In our CS (code-switch) dataset we applied some approaches as well as models to find out which one works better for this dataset. As we know, a dataset with mixed language is always critical to handle and for this reason, we tried to find out the best result from many ways.We used two approaches here:

1. Hybrid Sentiment Analysis
2. Fine tuning

In the first approach, we created lexicon-based prediction and model from the setup 'nlptown/bert-base-multilingual-uncased-sentiment' [32] based prediction. For these, we got their individual accuracy. Afterwards, for better accuracy, we created a hybrid model which had better results than both of them. In this part, we created a dataset containing Positive and Negative Bangla and English words. Then performed the preprocessing.
Following two processes, one for lexicon prediction and another for model prediction, we started our implementation. For lexicon, it removes non-alphabetic characters as well as all punctuation from the text ensuring only pure words are left. Then it converts all English words to lowercase except the Bangla to standardize the text and removes stop words of English and Bangla. Lastly, it splits the text into tokens. For the model prediction, the preprocess part is same as before, but the stop words are not removed and the text does not split into tokens. In the lexicon model, it counts the positive score and negative score (if the token matches with a word from

the positive column then +1 else -1). So from the condition– if a negative score > positive score then it returns negative, otherwise, it returns positive, and if it doesn't fall into any of the equations, it goes to Neutral. The model prediction uses the mBERT architecture [29] and predicts the sentiment of a text and returns with a star rating and confidence in this prediction. In instances where predictions yield a rating of either one or two stars, the outcome is classified as "Negative." A rating of three stars is classified "Neutral," while ratings above this threshold are categorized as "Positive." The hybrid model integrates both aforementioned approaches, stipulating that when model confidence surpasses 50%, the model's prediction is adopted. Conversely, if the confidence level is below this threshold, lexicon-based predictions are favored.Finally, the hybrid model decides the ultimate sentiment based on a very simple decision logic: if the lexicon-based sentiment is Neutral or the model's confidence is high, then it prefers the prediction from the model, otherwise uses the lexicon-based sentiment.
For the second approach, we have fine-tuned our dataset in the setup of 'FacebookAI/xlm-roberta-base'[17]. First, it preprocesses the text for both lexicon and model-based analyses. After pre-processing, it predicts sentiment using both types of sentiment analysis models.[20]

BERT:

Refining our sentiment analysis model based on the set of English-language reviews, we adopted a data-centric implementation to optimize the performance of the fine-tuned BERT model by considering factors such as token length and distribution. The setup of this model 'google-bert/bert-base-uncased' [10] involved using BERT's tokenizer in the pre-processing stage. Here, we filtered our dataset to the reviews having three or more words to ensure meaningful sentiment analysis. BERT's tokenizer analyzed the token length distribution. The maximum length of 128 tokens was selected as it was discovered that most of the samples were covered by the tokens whose length is not greater than 128, providing a good sequence length for the model shown in Figure 6.1.

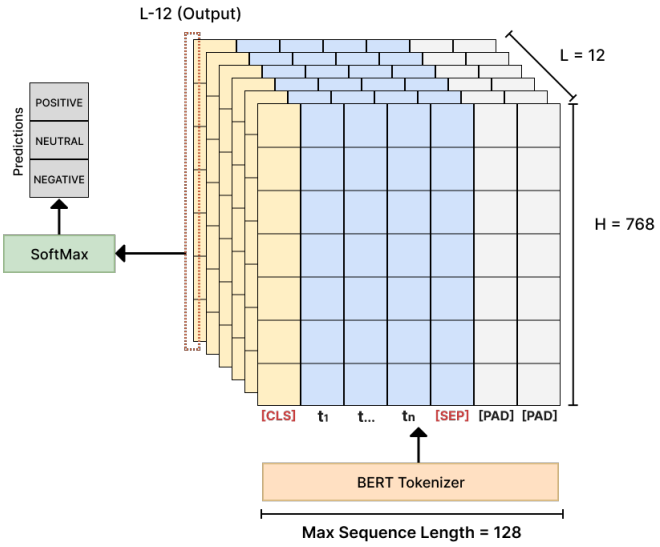


Figure 6.1: BERT fine-tuning for sentiment analysis

The selection of the 'bert-base-cased' model was made due to its medium size and for not being an overly complicated case of sentiment analysis in our research. The architecture was slightly modified, increasing the dropout rate from 0.25 to 0.3 to prevent overfitting and enhance generalization. To capture the sentiment of the entire review, the embedding of the [CLS] token was used. The insertion of special tokens is illustrated in Figure 6.2.

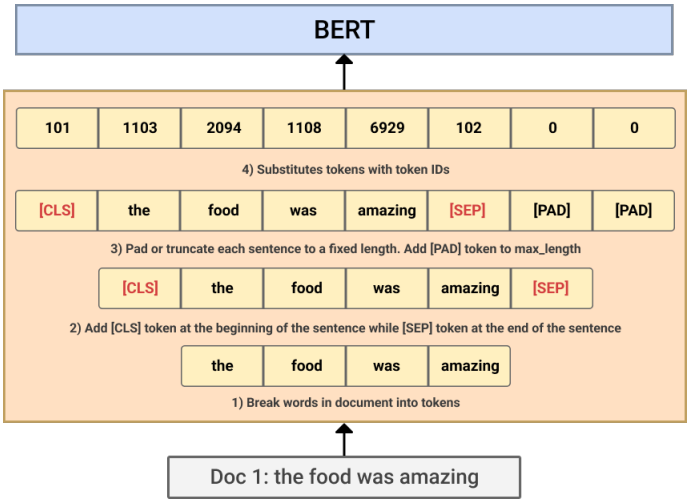


Figure 6.2: BERT Tokenizer: Insertion of Special Tokens

The hyperparameters of the model, specifically the learning rates were chosen after a series of experiments, the learning rates range from 2e-5 to 4e-5 to optimize the performance of the AdamW optimizer. An early stopping strategy was utilized with a patience setting of 5 epochs to track validation loss and prevent overfitting. This was an important stage in maintaining the model's generalization power. The aim of using 'bert-base-cased' with the restaurant reviews was to leverage BERT's powerful pre-trained context understanding and the cased model's cased-awareness property which is beneficial in sentiment analysis.

T5:

The Text-to-Text Transfer Transformer commonly referred to as T5. It is a Transformer-based framework that adopts a text-to-text methodology. In our English review dataset, we have used the T5 model to perform sentiment analysis. In the English dataset it has 47,398 reviews of restaurants, with each review assigned a rating between 1 and 5 Using this model we have predicted our sentiment labels (Positive, Negative, Neutral) for every review. Choosing this model was because of its success across a multitude of NLP challenges as well as its text-to-text framework. Also, the T5 model has shown its remarkable versatility across various tasks right from the start, thanks to its ability to adjust to different tasks simply by adding a task-specific prefix to the input sequence. At first, we loaded the dataset and preprocessed it. Then we divided the dataset into train-tests and after that, we initialized the SimpleT5 model and loaded the T5 base model which helps to fine-tune Sentiment analysis tasks and also numerous NLP tasks. Then the model trains the designated training dataset, where key parameters are: batch size, number of epochs, and token

lengths have been meticulously specified to optimize performance. The parameters of fine-tuning started with an examination of the lengths of the reviews, uncovering an average length of 15 words, and 95% of the reviews were 44 words or less. For this reason, source_max_token_len was set to 50 and target_max_token_len was set to 10 so that it could more accurately reflect the concise nature of sentiment labels. Also, 10 epochs were set to train the model with a batch size of 8. By this, the model fine-tunes the training dataset and evaluates it in the test dataset.

6.1.2 Long Short-Term Memory (LSTM):

In Natural Language Processing a type of RNN is called Long Short-Term Memory (LSTM), which can overcome the vanishing gradient problem associated with RNN. With LSTM, the model can determine which information is relevant in a text or paragraph and which is unnecessary. LSTM achieves this by using internal gates which are input gate, forget gate, and output gate and their respective formulas are:

$$\text{Input Gate, } I_t = \sigma\left(W_I\left[H_{t-1}, x_t\right] + b_F\right) \tag{6.1}$$

$$\text{Forget Gate, } F_t = \sigma\left(W_F\left[H_{t-1}, x_t\right] + b_F\right) \tag{6.2}$$

$$C'_t = \tanh\left(W_C\left[H_{t-1}, x_t\right] + b_C\right) \tag{6.3}$$

$$C_t = F_t * C_{t-1} + I_t * C'_t \tag{6.4}$$

$$H_t = O_t * \tanh\left(C_t\right) \tag{6.5}$$

$$\text{Output Gate, } O_t = \sigma\left(W_O\left[H_{t-1}, x_t\right] + b_O\right) \tag{6.6}$$

LSTM:

The model [2] was constructed with an embedding layer, followed by an LSTM layer of 256 units, a flatten layer, and a dense layer of 3 units and activation of softmax for three-class classification. The model was then configured for text data, providing a summary of insights into the architecture. Upon running the model, the output provided a summary of the model with an embedding layer of 3,400,800 parameters, an LSTM layer of 570,368 parameters, and the dense layer of 76,803 parameters. A total of 50 epochs was run for this model.

Attention-LSTM:

To make a successful use of the Attention LSTM [6] model we mapped the input sequence words, using an embedding layer, to 300-dimensional vectors. We followed this with an LSTM layer of 256 units. Consequently, we made use of TensorFlow and Keras, as the input sequences went through the embedding layer and then further processed through the LSTM layer. After adding the attention mechanism, a context vector was generated using the LSTM outputs' weighted sum. The attention-enhanced context output was concatenated with the original LSTM

output and flattened and passed through a dense layer of 3 units, and softmax activation for three-class classification. We compiled the model using the Adam optimizer and categorical cross-entropy loss function. Upon running the model, a summary of layers and output shape was derived to understand the process better. Showing that there are a total of 4,077,971 parameters, all of which are trainable. The performance was determined by running 50 epochs.

Biderectional-LSTM:

For implementing the Bi-LSTM [3], we construct a sequential model using an embedding layer with the input dimension being the vocabulary size of tokenized unique words. The structure of our sentiment analysis model included an embedding layer, followed by a Bidirectional LSTM layer with 256 units, a flatten layer, and a dense layer of 3 units with a softmax activation function for three-class classification. We implemented the Adam optimizer and categorical cross-entropy loss function to compile the model. The model was trained for 50 epochs.

6.2 Analyzing Sentiment in Review Text

We extended our analysis to include a textual sentiment evaluation, which was exclusively conducted on the text of the reviews, beyond the application of models to derive sentiment scores from reviews and ratings. Here, we did not remove the emojis from the texts so that we can get the predicted sentiment from the texts more accurately. We saw some discrepancies between the text's predicted sentiment with the rating's sentiment. It is because the user might have mistakenly given the rating to the platforms, while their review text shows a complete opposite sentiment. Table 6.1 depicts the similarity percentage of Predicted Sentiment with the Rating Sentiment.

Review Text	Sentiment Similarity
E	93.56%
B	90.71%
CM	56.73%
CS	74.0%

Table 6.1: Comparison of Sentiment Similarity Across Review Texts

Among the four dataset's, the CM review text showed a bit disparity when compared it's predicted sentiment of text with the rating sentiment. The reason behind this is discussed in the result analysis section. However, for the CM text's lower similarity percentage it's noisy and misspelled banglish representation can be another reason.

6.3 Result Analysis

For the Bangla dataset, we found that the BanglaBERT model's performance is similar to the other models that are being run on other datasets. According to the model description and analysis, we found that the BanglaBERT model gave us an accuracy of 78%, macro average F1 score of 54%, macro average recall of 58%, and

macro average precision of 83%. The low accuracy value can be credited to the discrepancy between the actual sentiment of the review and the star rating provided by the user. For instance, a very bad review having a 3-star rating and thus being labeled neutral when it should be negative. But considering the natural human language, the Bangla Bert can configure most of the sentiments correctly and efficiently.

As mentioned in the pre-processing section, we translate and transliterate CS and CM data respectively. Thus, the translated CS-B (completely Bangla CS reviews) and the transliterated Bangla script CM reviews were concatenated with the raw pure Bangla reviews and pre-processed together as a dataset containing about 11,000 entries. Then this dataset is fit into the BanglaBERT model to get an accuracy of 77%, macro average F1 score of 57%, recall of 59% and precision of 62%.

For the pure CM dataset evaluation using BanglishBERT model, we generated the accuracy, precision, recall and F1 score for each epoch. These results demonstrate a substantial increase in the accuracy and F1 score of the model during the training phase which highlights its adaptability and learning from the CM dataset. The results indicated and an initial training loss of 0.9516 that finally decreased to 0.0547 by the last epoch, which suggested successful learning. Nevertheless, validation loss grew from 0.7875 to 1.3380, indicating overfitting as the model learned. In spite of the overfitting, the model in the third epoch gave its best accuracy of 0.7734 and F1 score of 0.5918 indicating great sentiment classification. In the 10th epoch, the model achieved an accuracy of 76.09%, an F1 score of 66.80%, precision of 67.27%, and recall of 66.51%.Here, the precision and recall metrics keep showing consistent improvement that indicates a balanced identification of the sentiment classes.

For the English text, we began our work with LSTM models. Among those LSTM models, we found an accuracy of 73% using the Bi-LSTM model, 74% with Attention-LSTM and 74% with LSTM. And in case of precision among the LSTM models all three models showed a precision of 61%. It is also important to note that all three of the LSTM models were seen to return the lowest precision, recall and F1 score along with the accuracy rates. However, we were able to gain the most accuracy rates from the BERT model.

We fine-tuned the BERT uncased model for the English dataset. The training phase allows using the softmax activation function which produces a probability for each of the three sentiment classes that is needed for acquiring sentiment prediction. In particular, the softmax output from the [CLS] token indicated the model confidence level in all the classes at different stages of training.

$$\text{tensor}\left(\begin{bmatrix} 0.2932 & 0.1950 & 0.5118 \\ 0.1679 & 0.3118 & 0.5203 \\ \vdots & \vdots & \vdots \\ 0.2279 & 0.2444 & 0.5277 \end{bmatrix}, \text{device} = \texttt{'cuda:0'}, \text{grad\_fn} =< \text{SoftmaxBackward0} >\right)$$

The softmax layer outputs of the [CLS] token gave the probabilistic information regarding class predictions and kept on improving with each epoch giving a certain confidence of the model to its prediction. From the softmax outputs , we can predict that the model gives a probability of 29.32% to the 'negative' class, 19.50% to the 'neutral' class, and 51.18%

to the 'positive' class. In such a case, the model anticipates that the sentiment of the first example is 'positive' because that class has the highest probability. The increasing confidence shown by the softmax outputs over the epochs is a good marker of the model's performance improvement in sentiment class differentiation.

The training accuracy significantly increased with the initial value at 73.4% and peaking at 79.5% approximately. The validation curve had the same trend, beginning from 66.9% and increasing up to 78.9%. The trade of training loss drastically reduced from 0.747 and ended at approximately 0.556. The validation loss showed identical progress and reduced from around 0.850 to 0.571. From the accuracy curve shown in Figure 6.4, we can observe that with the convergence of the accuracy graph after initial fluctuation which indicates that the model was properly learning and validating patterns in data. The loss graph 6.3 exhibits a sharp training loss descent, which is a sign of learning and the validation loss also dropped showing that the model generalizes.
Another model used for the English dataset was T5. The T5 model exhibited strong capabilities in sentiment analysis, reflected by an F1 score of 0.643. This metric shows the model's proficiency in precisely identifying sentiments within a broad array of restaurant reviews. From the output, we got to know some changes which are:

- The `source_max_token_length` would remain at 50 as 95% of the reviews are 44 words.
- `target_max_token_len` can be reduced to 5 as the sentiment is short.
- As the reviews are short, the batch size can be increased to 16 or 32 for better performance.
- The dataloader workers can be increased up to 4 for in the fine-tune parameters, which will help the model gain more efficiency in future testing.

Model Names	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**
BanglaBERT (B)	78	83	58	83
BanglaBERT (B+CS+CM)	77	62	59	57
BanglishBERT (CM)	76	67	67	67
mBERT (CS)	71	59	64	60
XLM RoBERTa (CS)	77	59	59	59
BERT (E)	79	86	84	84
T5 (E)	70	70	70	64
Bi-LSTM (E)	73	61	60	61
Attention LSTM (E)	74	61	60	61
LSTM (E)	74	61	60	60

Table 6.2: Model Performance Metrics

The performance metrics for all of the models implemented on our four datasets are organised in Table 6.2.
.

6.3.1 Accuracy and Loss Curves:

Placing our best model's performance in visuals of loss and accuracy curves,side by side, gives us an adequate overview of these models. We can see that from among all the models, BERT and BanglaBERT performs well in terms loss and accuracy curves. While XLM-RoBERTa model shows a decent loss curve.

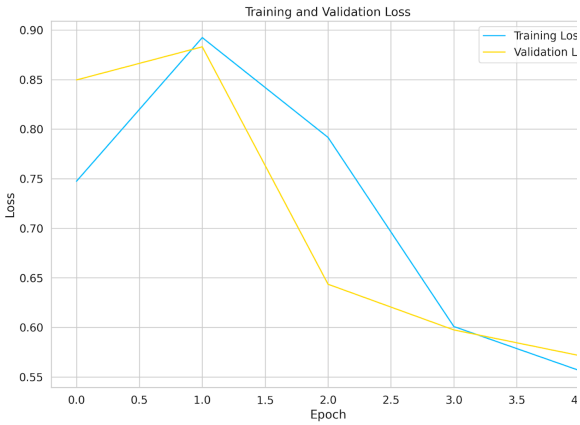


Figure 6.3: BERT Loss Curve

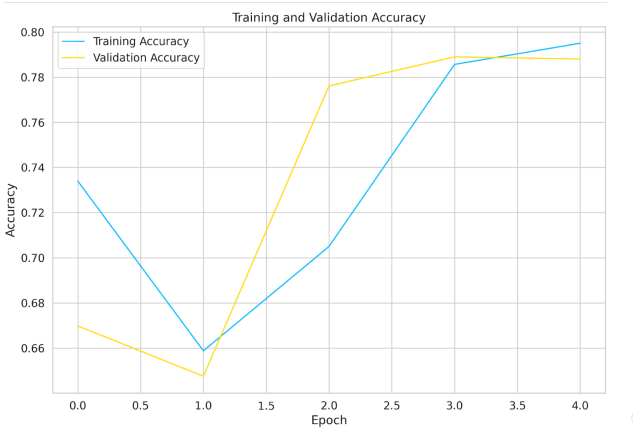


Figure 6.4: BERT accuracy Curve

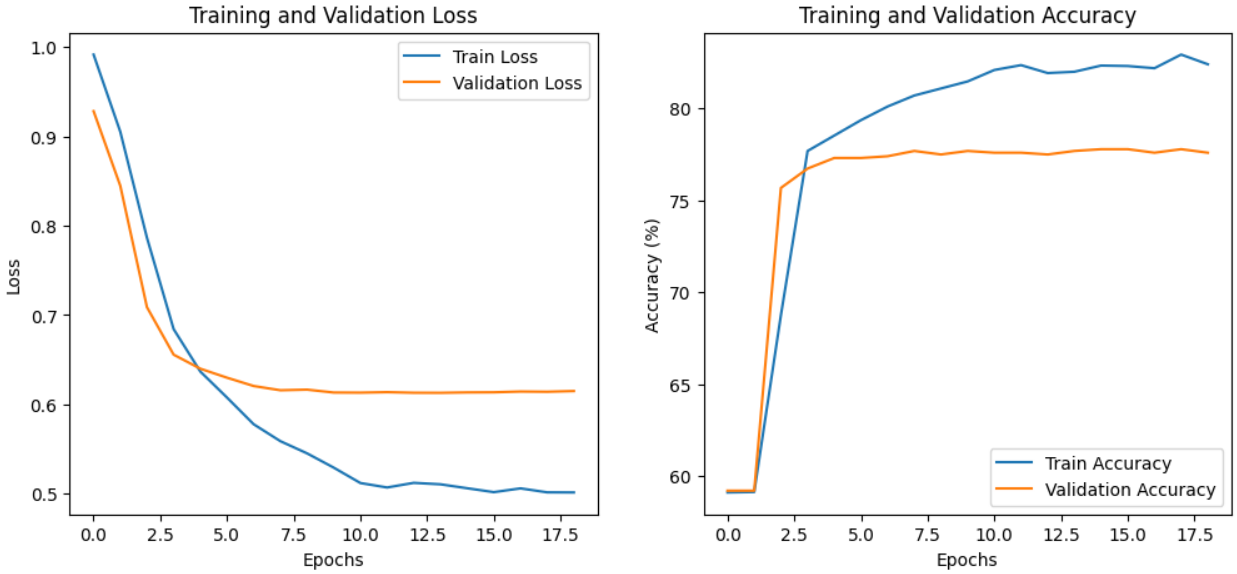


Figure 6.5: BanglaBERT Curve Representation on Bangla Dataset

.

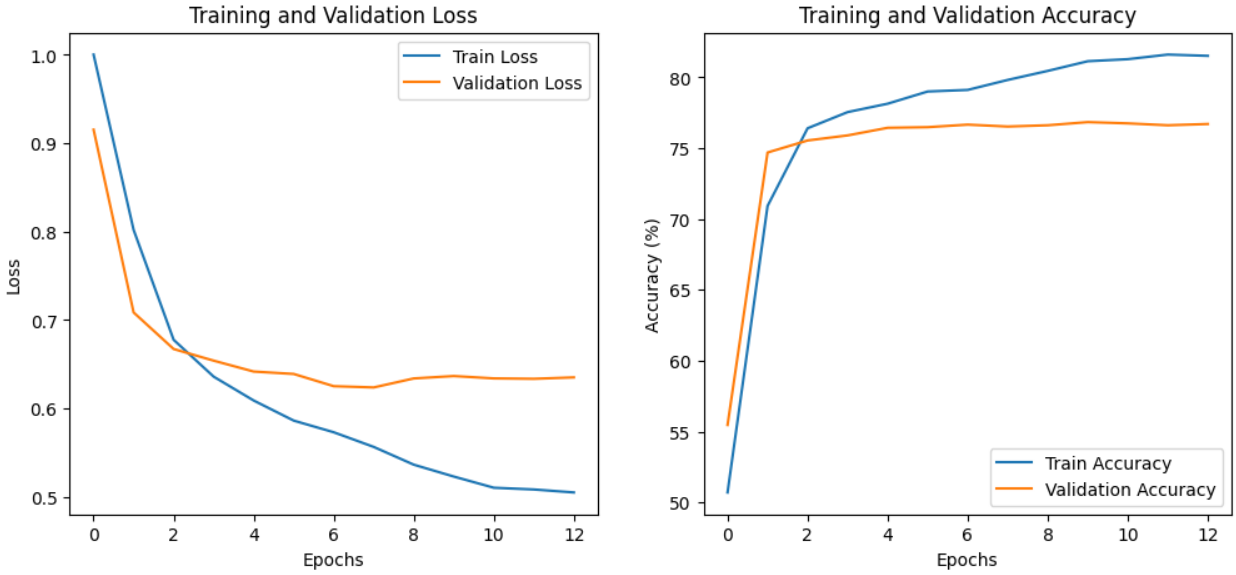


Figure 6.6: BanglaBERT Curve Representation on B+CS+CM Dataset

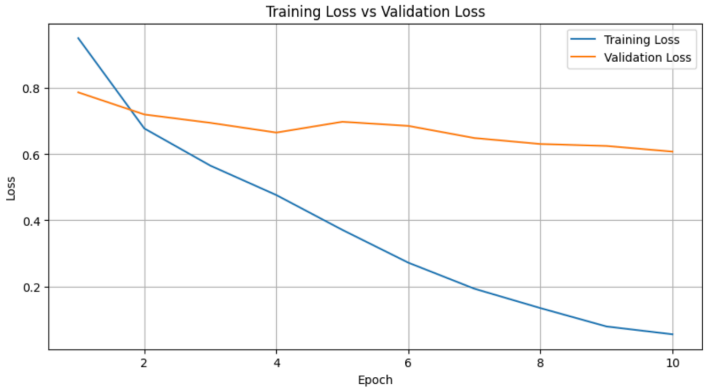


Figure 6.7: BanglishBERT Curve Representation on CM Dataset

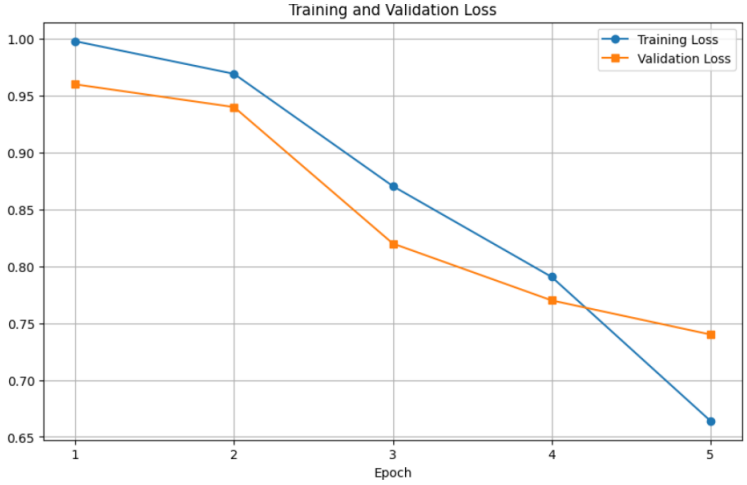


Figure 6.8: XLM-RoBERTa Curve Representation on CS Dataset

6.4 Clustering:

To identify which restaurants offered the same dishes or similar meals, we had to search through the user reviews of the particular restaurants. A single meal or food could be called different words depending on the region or dictionary. And a user wanting to search for food such as 'Pasta' might also be interested in foods like 'Spaghetti', 'Alfredo', 'Oven Baked Pasta' or 'Italian Pasta'. To solve this issue, we have decided to use try approach clustering based on food name prompts given by the user. In this paper, we have tested our clustering algorithm on three different model libraries. Namely spaCy, FastText, and all-MiniLM-L6-V2 from the Python sentence transformer library from Hugging Face.

spaCy:

Firstly, we tried using a medium-sized, pre-trained English language model from the "spaCy" library. This library had enough food words, and by using the similarity function, we could determine if the word was the same food as the given meal. To find our desired similar food, first, we concatenated all the normalized user reviews and used word extraction to keep the nouns and pronouns. We converted the list into a set, which helped to remove the duplicate words. As a result, our texts became shorter, and then we ran the similarity function on each of the words and measured the similarity score of the given keyword. When the similarity score exceeded a predefined threshold (established at 0.75 for this study), the corresponding term was collected. These similar terms were then located within our dataset to identify the names of restaurants associated with them. Although this methodology proved effective, it encountered drawbacks with local dishes not recognized by the spaCy English lexicon and with compound-named dishes (e.g., "French Fries," "Chicken Nuggets," "BBQ Chicken"). Additionally, the dataset included a significant amount of data in Banglish and Bangla, rendering the spaCy English Library ineffective for these entries. A further challenge was encountered with the similarity scoring mechanism, which occasionally gave high scores for distinct dishes, leading to classification errors even when the dishes were not identical.

All-MinLM-L6-V2:

Secondly, we tried to use SentenceTransformer models from the Hugging Face library and found that all-MiniLM-L6-V2 was the most effective and had faster computation time as it uses the Microsoft MiniLM architecture with only 6 layers. It is used for tasks such as clustering and semantic search. The input was normalized list of strings of the review column concatenated with the restaurant type to increase the size of the cluster per target word. We generated embeddings of our target food name word and the concatenated customer reviews. Then the dataset was split into batches, with a batch size of 1000 for effective use of memory and for each batch, the cosine similarity between the target embedding and the dataset embeddings was computed. If this similarity score passes a certain threshold then the index of the customer review is extracted and added to a new dataset. Since our dataset has four types of reviews, English, Bangla, CM and CS, thus the dataset was separated by language and each language had a specific similarity score threshold assigned to it. So for English and CM, it was 0.5, for Bangla it was 0.8, and as CS contained both Bangla and English words thus the CS dataset went through the model twice. Once with an English target word and then again with a Bangla target word and thus had a similarity threshold according to the target word. It was seen that

this model worked extremely well when clustering Bangla, CM, and CS reviews based on similar words as the target when compared to the other models. However, the top results of all-MiniLM-L6-V2 clustering reviews were mostly words with similar spellings, and words with semantic similarity were fewer than what we wanted. Thus when clustering using the target word "bhaji", the top results are variations of the word bhaji instead of semantically similar words such as 'shobji' or 'fish'.

FastText:

Our primary goal was to extract words that were semantically similar to the target word. In this way, we could achieve the main word as well as relevant words associated with it. FastText library, which is mainly used for text classification tasks as well as word similarity calculation and extraction, came in handy while performing this task of similar word extraction. We used an unsupervised training approach to train the Fasttext model on normalized words for it to learn word vectors. This enabled us to get words that were semantically similar to our input based on context. Since we have 4 types of languages, we had to split our dataset according to language type, clean it, and pass it down as 4 different models. The similar words that were found using the 'get_nearest_neighbors()' function proved to be closely related to our target word. The main idea behind calculating similarity scores leveraged the concept of Cosine Similarity calculation shown in equation 6.7.

$$\text{Cosine similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{6.7}$$

Other necessary parameters such as learning rate, wordNgrames, etc. were adjusted per model based on the particular portion of dataset passed. While the model that was trained on pure English data returned an impressive result irrespective of the target food type being Western or Bengali, the Bangla, CM, and CS models needed adjusting the parameters. If the target word's origin was Bengali like 'bhaji' transcribed in Bangla, then these models would work well, otherwise for non-Bengali food items, the result was somewhat inadequate in comparison to all-MiniLM-L6-V2. The possible reasons for this were due to the individual Bangla, CM, and CS data frame dimensions being insignificant in comparison to pure English data. Another reason included the presence of misspelled words present in the dataset which occurred in CM in particular and were not identified as noise even after performing normalization. This is because people's way of writing words in Banglish varies, and do not always follow a particular set of rules which combines typos, their way of writing (such as 'bhajichilo', 'bhaji', 'vaji') etc. While all-MiniLM-L6-V2 captured syntactically similar words, Fasttext focused more on the semantic analysis which resulted in this outcome as depicted in figure 6.9.

Target Word	Language		Model					
			Fasttext		all-MiniLM-L6-v2		SpaCy	
			Similar words	Similarity score	Similar words	Similarity score	Similar words	Similarity score
Pizza	English (Pizza)		Pizza	1	pizza	1	pizza	1
			chizza	0.884172	pizza restaurant	0.759611	pizzaz	0.890376
			pizzaburg	0.835759	pizzawas	0.739289	sandwich	0.768141
			meatzza	0.79622	pizzasvery	0.722643	burger	0.752919
			crusted	0.787955	pizzasize	0.721	cheeseburger	0.752718
	Bangla (পিজ্জা)		পিজ্জা	1	পিজ্জা	1.0000002	N/A	N/A
			পিজ্জাটা	0.924358	পিজজা	1	N/A	N/A
			পিজা	0.89716	পিজ্জায়	0.986724	N/A	N/A
			পিজাটা	0.859845	পিজ্জ্বায়	0.986724	N/A	N/A
			টপিং	0.734184	পিজা	0.979337	N/A	N/A
	Code Mix (CM) (Pizza)		Pizza	1	pizza	1	N/A	N/A
			pizzaburg	0.999975	pizzas	0.900104	N/A	N/A
			tomato	0.999962	pizza restaurant	0.759611	N/A	N/A
			pasta	0.99996	pizzay	0.732789	N/A	N/A
			chesse	0.999958	pizzao	0.72524	N/A	N/A
	Code Switch (CS) (Pizza/ পিজা)	English (Pizza)	Pizza	1	pizza	1	N/A	N/A
			pizza	0.999985	pizza restaurant	0.759611	N/A	N/A
			Pizza	0.999981	pizza chain	0.709255	N/A	N/A
			total	0.999981	food	0.637525	N/A	N/A
			Chatime	0.99998	pizzarolla	0.63031	N/A	N/A
		Bangla (পিজ্জা)	পিজ্জা	1	পিজ্জা	1	N/A	N/A
			পিজ্জা,	0.999988	পিজ্জ্যা	0.983302	N/A	N/A
			পিজ্জার	0.999987	পিজা	0.979337	N/A	N/A
			নিয়েছি	0.999971	পিজ্জার	0.979146	N/A	N/A
			মাশরুম	0.999971	পিজ্জাটা	0.969386	N/A	N/A
Bhaji	English (Bhaji)		Bhaji	1	bhaji	1	bhaji	1
			vaji	0.945138	bhajis	0.89485	haji	0.9011178
			bhat	0.930166	baji	0.827276	pav	0.7874076
			bhaat	0.928845	bhaaji	0.776195	halwa	0.7839658
			bharta	0.923365	bheja	0.758076	sabji	0.7838063
	Bangla (ভাজি)		ভাজি	1	ভাজি	1	N/A	N/A
			ভর্তাটা	0.999762	ভাজির	0.974269	N/A	N/A
			ডাল	0.999706	ভাজে	0.966772	N/A	N/A
			শুটকি	0.999679	ভাজিটা	0.965656	N/A	N/A
			ভর্তি	0.99966	ভাইজা	0.963846	N/A	N/A
	Code Mix (CM) (Bhaji)		Bhaji	1	bhaji	1	N/A	N/A
			bhaja	0.998615	bhajis	0.89485	N/A	N/A
			alur	0.99699	baji	0.827276	N/A	N/A
			alu	0.99595	buji	0.792383	N/A	N/A
			begun	0.995937	bhujha	0.791867	N/A	N/A
	Code Switch (CS) (Bhaji/ ভাজি)	English (Bhaji)	Bhaji	1	bhaji	1	N/A	N/A
			aloo	0.997373	bhalo	0.693017	N/A	N/A
			dim	0.997368	bhai	0.679451	N/A	N/A
			bhorta	0.997324	bhuna	0.657181	N/A	N/A
			rice	0.997147	vaji	0.624946	N/A	N/A
		Bangla (ভাজি)	ভাজি	1	ভাজি	1	N/A	N/A
			ভাজা	0.868506	ভাজিটা	0.965656	N/A	N/A
			লবণ	0.846988	ভাঁজা	0.963709	N/A	N/A
			মাছ	0.829993	ভোজ	0.959712	N/A	N/A
			চুই	0.816306	ভোজন	0.946617	N/A	N/A

Figure 6.9: Similarity Scores Comparison Between Models

Chapter 7

Recommendation System

The advancement of the incorporation of sentiment analysis and textual similarity measures strengthens restaurant recommender systems that depend on user-generated content. This approach consists of several computational steps based on sentiment analysis and machine learning techniques to understand the sentiment expressed in reviews and measure it in a way useful for the recommendation.

This paper focuses on the strategic fusion of sentiment-weighted scoring systems and similarity metrics that considerably refine the recommendations. This method integrates semantic information with the content of user reviews using FastText for semantic similarity and TF-IDF vectors for content specificity. The last similarity measure, named the Combined Similarity Score, merges these approaches to enhance the system's performance of providing relevant recommendations.

7.1 Sentiment Analysis and Popularity Score

The system's base is sentiment analysis which is especially based on the Bert model having 79% accuracy for the 'E' annotated dataset based on the Language column and the BangalBert model having 78% accuracy for the 'B' annotated dataset. In this process, the predicted sentiments of the Review column are classified into positive, neutral, and negative sentiments with values of +1, 0, and -1, respectively. The values are then normalized by the accuracy of the model to obtain a Weighted Sentiment Score for each review. The average of these scores at the restaurant level gives us a measure of the overall sentiment towards each establishment. The mean and standard deviation of user ratings are used to construct the Popularity Score To complement the sentiment analysis. This score measures a restaurant's popularity among the customers, serving as a significant factor in the recommendation system.

The initial Overall Sentiment Score is determined by calculating the average weighted sentiment score, user ratings, and the following calculation of the Combined Similarity Score:

$$\text{Overall Sentiment Score} = \frac{\text{Weighted Sentiment Score (average)} + \text{Combined Similarity} + \text{Rating}}{3} \tag{7.1}$$

Sentiment Value Mapping:

$$\begin{cases} 1 & \text{if Predicted Sentiment } = 2 \text{ (Positive)} \\ 0 & \text{if Predicted Sentiment } = 1 \text{ (Neutral)} \\ -1 & \text{if Predicted Sentiment } = 0 \text{ (Negative)} \end{cases}$$

Weighted Sentiment Score Calculation:

Weighted Sentiment Score = Sentiment Value × Model Accuracy

It is to mention that, Predicted Sentiment is the sentiment analysis model's output of a review that falls into Positive (2), Neutral (1), or Negative (0). Sentiment Value simply converts the rendered sentiment into a numeric value, which in turn allows for quantitative analysis. Model Accuracy is a measure of how reliable the sentiment analysis model is in the prediction of sentiments.

7.2 Methodology for Combined Similarity Score

Semantic Similarity via FastText

First, FastText embeddings are utilized by the system to represent the semantic essence of user-generated reviews. FastText creates embeddings for each review by averaging vector word representations. Then, cosine similarity between the pair of reviews provides a semantic similarity score, which is the measure of semantic proximity.

Content-Specific Similarity via TF-IDF

In addition to the semantic analysis, TF-IDF vectors are created for the reviews. This approach emphasizes the unique terms of each document and in such a manner helps calculate content-based similarity, which is very useful in the recommendation process.

Integrating Similarity Measures

The Combined Similarity Score is derived by taking the weighted average of the normalized FastText and TF-IDF similarity scores:

Combined Similarity = $\alpha\times$ Normalized FastText Similarity $+\beta\times$ Normalized TF-IDF Similarity

Here, α and β represent the weights given to semantic and context-specific similarities, respectively. This integration effectively addresses the substantial semantic connections with the informational elements of the content presented in reviews. In this case, the weights attached were as follows–

$$\text{weight\_fasttext}(\alpha) = 0.7$$

$$\text{weight\_tfidf}(\beta) = 0.3$$

Semantic similarity was given the highest consideration in the recommender system which is focused on understanding sentiments and thematic topics in user reviews. To account

for this, we changed the values of the weights to 0.7 for FastText similarity, which captures semantic differences, and 0.3 for TF-IDF similarity, that is content depictions. This determination aligns with our goal of providing recommendations that deeply associate with users based on their sentiments.

7.3 Adjusted Overall Sentiment Score

Recognizing the need for a more balanced approach and depending on the actual influence of every component, the Adjusted Overall Sentiment Score is proposed. This modification normalizes the Rating to a scale of -1 to 1, matching it with the sentiment value scale, and assigning different weights to the normalized rating, weighted sentiment score, and combined similarity score. This revised score attempts to represent user preferences more accurately by favoring explicit user feedback through ratings, while also benefiting from the refined insights of sentiment analysis and textual similarity:

$$\begin{aligned}\text{Adjusted Overall Sentiment Score (AOSS)} = &(\text{Normalized Rating} \times 0.5) \\ &+ (\text{Weighted Sentiment Score (average)} \times 0.25) \\ &+ (\text{Combined Similarity} \times 0.25) \end{aligned} \tag{7.2}$$

The Adjusted Overall Sentiment Score given in equation 7.2 is an effective approach for conducting restaurant evaluations and recommendations. Because it incorporates user ratings, sentiment analysis, and similarity measures to provide an in-depth evaluation.

Weight Configuration of AOSS:

Here, choosing the weight configuration where the split setting was as follows: Rating Weight: 0.5, Sentiment Weight: 0.25, Similarity Weight: 0.25 is considered more appropriate mainly as a concession to the complexity of the user's preference, unlike the high rating weight configurations that achieved 100% accuracy in some matrices (when given more than 50% weight to Star Rating given by user). Because of this complexity, overfitting will not be able to capture other more subtle features of sentiment analysis and textual similarity, in other words, the loss of a more diverse set of users preference. In this context, a balanced weighted approach eliminates some of the user ratings innate biases, allows to prolong users' engagement by content browsing and turns the system into a powerful, fair recommendation engine that closely emulates individual user preference based on the text (Review) feedback provided, giving the users with a rounded, intriguing experience. The empirical validity of such configuration supported by A/B testing shows that the best possible relevance of the recommendations, and user satisfaction is a better option than the tempting perfect scores in the less balanced configurations.

Rating Weight	**Sentiment Weight**	**Similarity Weight**	**Precision**	**Recall**	**F1 Score**
0.5	0.25	0.25	1.0	0.9809	0.9903
0.3	0.4	0.3	0.9231	0.9651	0.9436
0.3	0.3	0.4	0.9231	0.9651	0.9436
0.25	0.5	0.25	0.9187	0.9084	0.9135
0.25	0.25	0.5	0.9232	0.9670	0.9446
0.2	0.6	0.4	0.8848	0.9420	0.9125
0.2	0.2	0.6	0.8866	0.9789	0.9305

Table 7.1: Comparative Analysis of Weight Configurations for the AOSS

In 7.1, the table provides a systematic comparison of how different weight configurations perform for rating, sentiment, and similarity in a restaurant recommender system. The configuration selected, 0.5, 0.25, 0.25 for Rating Weight, Sentiment Weight, Similarity Weight shows the highest F1 Score of 0.9903, which is a maximum balance between precision and recall. Therefore, an intermediate approach, as opposed to over-weighting of one component, improves the success of recommendations provided by the system.

Application to Recommender System:

The restaurant recommender system's model incorporates sentiment analysis to identify user preferences and sentiment polarity in reviews. Advanced statistical techniques generate recommendations that align with user preferences through fine-grained detection of textual similarities, ensuring a seamless experience. This represents a significant advancement in recommender systems, incorporating sentiment analysis and semantic complexity, thereby improving user experience.

7.4 Result Analysis

In the analysis provided, the Adjusted Overall Sentiment Score is used as a measure of consumer sentiment towards different restaurants, and the score is normalized on a scale in which a higher score indicates a more favorable consumer sentiment.

Identification and Ranking of Top Recommended Restaurants:

For measuring the efficiency of our recommender system, we worked out an algorithm that provides a comprehensive list of all the restaurants and prioritizes them based on their AOSS. This metric is calculated by combining three different factors: user ratings, sentiment analysis of user reviews, and text semantic similarity metrics. The measure of the impact of each factor on the overall metrics is by the weight with the highest KPI as discussed in the previous section. Here, these weights set a balance between the input from users and the valuable insights provided by review content.

Here, our methodology was to rank the restaurants directly by their Adjusted Overall Sentiment Score. This procedure improves the recommendation process by highlighting the highest scoring restaurant as the top recommendations.

This methodology has led us to identify the top five restaurants showcasing the highest AOSS's within four splitted dataset (E, B, CM, CS). Table 7.2 shows these establishments in descending order based on their scores for the restaurants contained in English dataset.

Restaurant Name	**AOSS**
The Pabulum	0.922461
Madchef	0.920041
Burger King	0.918998
Takeout	0.916942
Pizza Lab	0.915507

Table 7.2: Restaurants with highest AOSS from English Dataset

Similarly, Table 7.3 shows the top 5 restaurants with highest AOSS from the restaurants of our Bangla dataset.

Restaurant Name	**AOSS**
সিরাজ চুইগোস্ত	0.934996
Vorta Vaji Resataurant	0.928412
Sat Sotero Restaurant	0.927866
Hawlader Biriyani House	0.927857
Bismillah Hotel Restaurant & Kabab Ghar	0.927857

Table 7.3: Restaurants with highest AOSS from Bangla Dataset

It's noteworthy to mention that from Table 7.4 and 7.5, we found the similiar restaurant recommendations even though their AOSS's are different across their respective datasets.

Restaurant Name	**AOSS**
Redres	0.931097
Wafas Kabab Fort	0.921801
Jhawbon Classic Restaurant	0.914277
Cheese's Crushed	0.911152
বাঙালিয়ানা ভোজ	0.906780

Table 7.4: Restaurants with highest AOSS from CM Dataset

Restaurant Name	**AOSS**
Redres	0.936119
Wafas Kabab Fort	0.926828
Jhawbon Classic Restaurant	0.919283
Cheese's Crushed	0.916196
বাঙালিয়ানা ভোজ	0.911801

Table 7.5: Restaurants with highest AOSS from CS Dataset

The above mentioned table outputs list the five best restaurants across four datasets, that satisfy the recommendation and liking criterion by having the highest Adjusted Overall Sentient Scores (AOSS). This method offers an operative and systemized way of determining the best restaurants through customer judgment and could serve as a good recommendation system.

Chapter 8

Limitations

The Need for Constant Dataset Update:
With every passing day, new reviews are constantly being added under restaurant name labels. This means that the current review data that we hold will need to be updated too. Considering extreme cases, a restaurant that initially had a poor review might even start getting positive reviews as days progress. Therefore this old data might prove inaccurate and even misleading.

Standardization makes it Unfair for Bigger Restaurants:
Larger and popular restaurant chains usually contain reviews crossing a five thousand review mark. And since we extracted only a fraction of those data, the information contained might not properly portray the real scenario. After standardizing, the value falls down even more, giving rise to a possible adverse effect.

Language Bias:
The fact that the number of English data encompass nearly 80% of the dataset, creates a possible language bias. Bringing uniformity to the data ratio, although challenging, might help draw a uniform conclusion to the results.

Single-outlet Restaurants face an Uneven Competition:
Smaller and less popular restaurants fall under the shade of an uneven competition due to having fewer or no outlets at all. Thus these restaurants contain less reviews which makes

Chapter 9

Future Work

Input-Output Based Recommendation:
Our future word will include searching for similar food names along with the location. The output will show the recommended restaurants that contain similar food names within that location with AOSS(Adjusted Overall Sentiment Score) so that the user can understand which restaurant is recommended or not based on the AOSS.

Continuation of Dataset:
We are aimed to constantly update our dataset to keep updated with the changing sentiments of the restaurants.

Work on Language Bias:
As the percentage ratio of Bangla, CS and CM reviews to English is uneven, where the English data is predominant holding almost 80% reviews of entire dataset, we will work on increasing the Bangla,CS and CM reviews.

Chapter 10

Conclusion

Amid the chaos of our daily lives, to find some calm, relax, hang out, or have some good meals together, we visit multiple restaurants and try out different cuisines. However, the accessibility and number of restaurants have become so enormous that we are often confused about where to find our desired place with our preferred cuisine. To resolve this problem, restaurant and cuisine recommendations according to the user's interests are badly needed. To do that, this proposed research might create an impact to ensure maximum customer satisfaction. Besides that, as we are recommending restaurants to respective customers' needs specifically thus, it is most likely that we will have those customers repeatedly. This might also help the restaurants' acquire their target customers effectively and efficiently. Considering both factors and analyzing the correlation of both sides might give us a promising recommendation. As our project focuses on urban communities, this might conflict with the trends of other regions. Moreover, our research is based on actual raw data from platforms that ended up with a significant English-speaking user base, so we found insufficient data in all the other languages like Bangla, Code Switch (Bangla and English), and Code Mix (Banglish). Despite that, our proposed research has managed to generate an efficient recommendation system by integrating sentiment score with textual similarity.

Bibliography

[1] J. A. Czepiel, M. R. Solomon, C. F. Surprenant, and E. G. Gutman, "'service encounters: An overview'," *The service encounter: Managing employee/customer interaction in service businesses*, pp. 3–15, 1985.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997. DOI: 10.1109/78.650093.

[4] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," *arXiv preprint cs/0409058*, 2004.

[5] M. Neethu and R. Rajasree, "Sentiment analysis in twitter using machine learning techniques," in *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)*, IEEE, 2013, pp. 1–5.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014. [Online]. Available: https://arxiv.org/abs/1409.0473.

[7] G. Gautam and D. Yadav, "Sentiment analysis of twitter data using machine learning approaches and semantic analysis," in *2014 Seventh international conference on contemporary computing (IC3)*, IEEE, 2014, pp. 437–442.

[8] A. P. Jain and P. Dandannavar, "Application of machine learning techniques to sentiment analysis," in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, IEEE, 2016, pp. 628–632.

[9] Fastfoodbd, *Fast food culture in bangladesh*, Jan. 2017. [Online]. Available: https://fastfoodbd.wordpress.com/2017/01/12/fast-food-culture-in-bangladesh/.

[10] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. arXiv: 1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805.

[11] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," *arXiv preprint arXiv:1802.06893*, 2018.

[12] S. Mandal, S. K. Mahata, and D. Das, "Preparing bengali-english code-mixed corpus for sentiment analysis of indian languages," *arXiv preprint arXiv:1803.04000*, 2018.

[13] S. Willas, *7 reasons why online reviews are essential for your brand*, Sep. 2018. [Online]. Available: https://mention.com/en/blog/online-reviews/.

[14] A. Krishna, V. Akhilesh, A. Aich, and C. Hegde, "Sentiment analysis of restaurant reviews using machine learning techniques," in *Emerging Research in Electronics, Computer Science and Technology: Proceedings of International Conference, ICERECT 2018*, Springer, 2019, pp. 687–696.

[15] S. Hu, A. Kumar, F. Al-Turjman, S. Gupta, S. Seth, *et al.*, "Reviewer credibility and sentiment analysis based user profile modelling for online product recommendation," *IEEE Access*, vol. 8, pp. 26 172–26 189, 2020.

[16] L. Kohoutová, J. Heo, S. Cha, *et al.*, "Toward a unified framework for interpreting machine-learning models in neuroimaging," *Nature protocols*, vol. 15, no. 4, pp. 1399–1435, 2020.

[17] A. Younas, R. Nasim, S. Ali, G. Wang, and F. Qi, "Sentiment analysis of code-mixed roman urdu-english social media text using deep learning approaches," in *2020 IEEE 23rd International Conference on Computational Science and Engineering (CSE)*, IEEE, 2020, pp. 66–71.

[18] E. Amparore, A. Perotti, and P. Bajardi, "To trust or not to trust an explanation: Using leaf to evaluate local linear xai methods," *PeerJ Computer Science*, vol. 7, e479, 2021.

[19] E. Asani, H. Vahdat-Nejad, and J. Sadri, "Restaurant recommender system based on sentiment analysis," *Machine Learning with Applications*, vol. 6, p. 100 114, 2021.

[20] F. Barbieri, L. E. Anke, and J. Camacho-Collados, "Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond," *arXiv preprint arXiv:2104.12250*, 2021.

[21] K. I. Islam, S. Kar, M. S. Islam, and M. R. Amin, "SentNoB: A dataset for analysing sentiment on noisy Bangla texts," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3265–3271. DOI: 10.18653/v1/2021.findings-emnlp.278. [Online]. Available: https://aclanthology.org/2021.findings-emnlp.278.

[22] H. Khatter, N. Goel, N. Gupta, and M. Gulati, "Movie recommendation system using cosine similarity with sentiment analysis," in *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, IEEE, 2021, pp. 597–603.

[23] N. A. Osman, S. A. Mohd Noah, M. Darwich, and M. Mohd, "Integrating contextual sentiment analysis in collaborative recommender systems," *Plos one*, vol. 16, no. 3, e0248695, 2021.

[24] A. Adak, B. Pradhan, N. Shukla, and A. Alamri, "Unboxing deep learning model of food delivery service reviews using explainable artificial intelligence (xai) technique," *Foods*, vol. 11, no. 14, p. 2019, 2022.

[25] A. Bhattacharjee, T. Hasan, W. Ahmad, *et al.*, "BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla," in *Findings of the Association for Computational Linguistics: NAACL 2022*, Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 1318–1327. [Online]. Available: https://aclanthology.org/2022.findings-naacl.98.

[26] S. Chowdhury, *Evaluating cold-start in recommendation systems using a hybrid model based on factorization machines and sbert embeddings*, 2022.

[27] P. Ni, Y. Li, and V. Chang, "Recommendation and sentiment analysis based on consumer review and rating," in *Research Anthology on Implementing Sentiment Analysis Across Multiple Disciplines*, IGI Global, 2022, pp. 1633–1649.

[28] I. Ramadasa, L. Liyanage, P. Asanka, and T. Dilanka, "Analysis of the effectiveness of using google translations api for nlp of sinhalese," Sep. 2022.

[29] F. D. Souza and J. B. d. O. e. S. Filho, "Bert for sentiment analysis: Pre-trained and fine-tuned alternatives," in *International Conference on Computational Processing of the Portuguese Language*, Springer, 2022, pp. 209–218.

[30] D. Tribune, "Hospitality and the restaurant sectors grows 58% in 11 years," *Dhaka Tribune*, Dec. 2022. [Online]. Available: https://www.dhakatribune.com/business/2022/12/07/hospitality-and-the-restaurant-sectors-grows-58-in-11-years.

[31] C. Farrelly, Y. Singh, Q. Hathaway, *et al.*, "Current topological and machine learning applications for bias detection in text," Nov. 2023. DOI: 10.13140/RG.2.2.22623.43680.

[32] N. Town, *Bert-base-multilingual-uncased-sentiment*, https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment, Accessed: 2023-04-03, 2023.

[33] T. Singh, A. Raut, D. Agarwal, R. Jha, A. Rai, and M. Kumar, "Food recommendation system using neural collaborative filtering and sentiment analysis,"