

UNIVERSITY OF TARTU  
Institute of Computer Science  
Computer Science Curriculum

**Anton Berik**  
**AI-Powered Food Safari:**  
**Restaurant Picks for the Modern Traveller**  
**Bachelor's Thesis (9 ECTS)**

Supervisor:  
Somnath Banerjee, PhD

# AI-Powered Food Safari: Restaurant Picks for the Modern Traveller

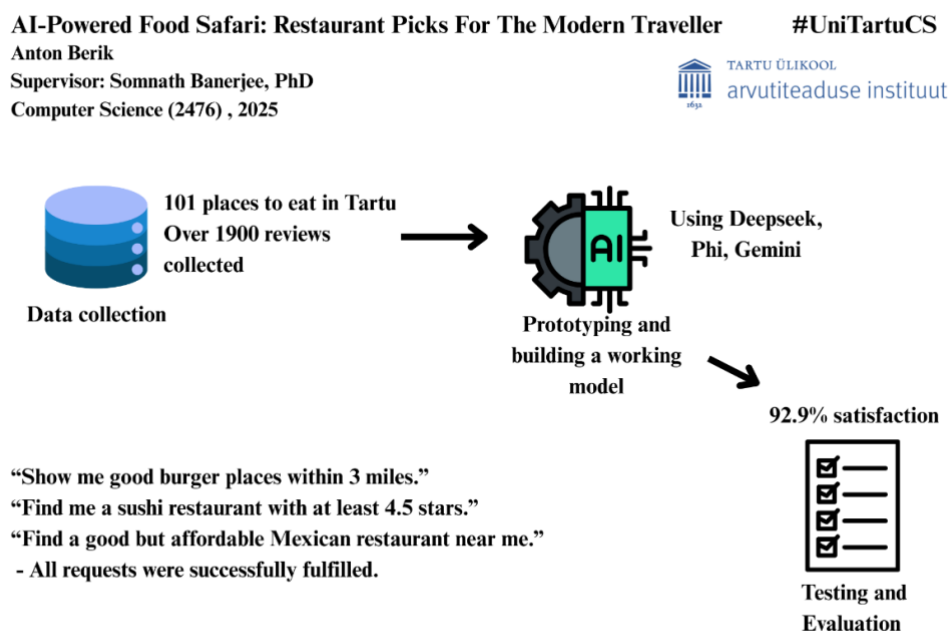
## Abstract:

This thesis presents a system for AI-powered restaurant recommendation based on user preferences, customer reviews, and metadata. The approach leverages natural language processing techniques, in particular, aspect-based sentiment analysis and Large Language Model evaluate restaurants on a wide range of factors, such as food quality, service, and atmosphere. By interpreting user prompts, the system adjusts the importance of different aspects and uses sentiment scores from reviews alongside restaurant metadata like location and ratings to calculate a final ranking. The system supports multilingual inputs and includes steps for data collection, cleaning, and structured analysis. The result is a personalized and flexible recommendation engine that provides users with relevant and context-aware restaurant suggestions.

**Keywords:** Aspect-Based Sentiment Analysis, Natural Language Processing, Recommendation Systems, Restaurant Ranking, User Preference Modeling

**CERCS:** P176 – Artificial intelligence

## Visual Abstract:



# Tehisintellekti Põhjal Toimiv Toiduretk: Restoranisoovitused Kaasaegsele Reisijale

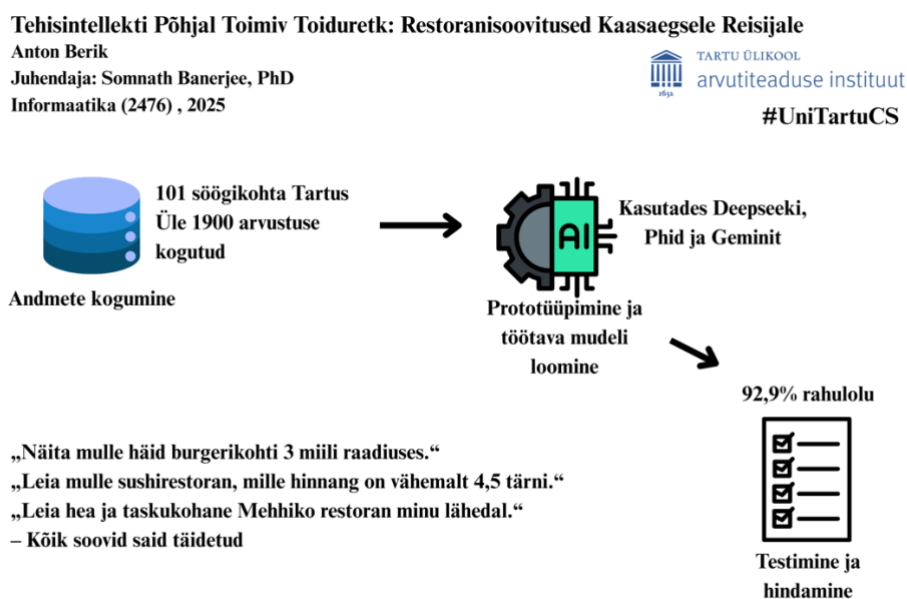
## Lühikokkuvõte:

Käesolev lõputöö tutvustab tehisintellektil põhinevat restoranide soovitusüsteemi, mis tugineb kasutaja eelistustele, klientide arvustustele ja metaandmetele. Lähenemine kasutab loomuliku keele töötlemise tehnikaid, eelkõige, aspektipõhise sentimentanalüüsi ja suuri keelemudeleid, et hinnata restorane mitmesuguste tegurite alusel, nagu toidu kvaliteet, teenindus ja õhkkond. Süsteem tõlgendab kasutajate sisendeid ning kohandab aspektide olulisust vastavalt, kasutades arvustustest saadud hinnanguid koos restorani metaandmetega (nt asukoht, reitingud) lõpliku paremusjärjestuse koostamiseks. Süsteem toetab mitmekeelseid sisendeid ja koosneb sammudest andmete kogumiseks, puhastamiseks ning struktureeritud analüüsiks. Tulemuseks on isikupärastatud ja paindlik soovitusmootor, mis pakub kasutajale kontekstitundlikke ja asjakohaseid restorani soovitusi.

**Märksõnad:** Aspektipõhine sentimentanalüüs, Loomuliku Keele Töötlus, Soovitusüsteemid, Restoranide Järjestamine, Kasutajate Eelistuste Modelleerimine

**CERCS:** P176 – Tehisintellekt

## Visuaalne Lühikokkuvõte:



# Table of Contents

<b>1. Introduction</b>	<b>5</b>
<b>2. Background</b>	<b>6</b>
2.1 Natural Language Processing in Review Analysis	6
2.2 Data Mining	7
2.3 Prompt Engineering	8
2.3.1 Introduction to Prompt Engineering	8
2.3.2 Prompt Types and Techniques	9
2.4 Related Works	9
<b>3. Method and System Development</b>	<b>11</b>
3.1 Data Collection	11
3.2 Evaluation: Use cases	12
3.3 LLM-based Aspect Extraction	13
3.4 AI-assisted Recommendations	14
3.4.1 Ranking Model	14
3.4.2 Template Model Development	16
3.4.3 Prompt Template Description for Recommendation	16
<b>4. Evaluation and Results</b>	<b>21</b>
4.1 Evaluation Setup	21
4.2 Evaluation Metric: Top-5 Accuracy	22
4.3 Results	22
4.4 Demonstration of 2 Use Cases:	23
<b>5. Implementation and Further Developments</b>	<b>26</b>
5.1 Mobile Application	26
5.2 Web Application	26
<b>6. Conclusion</b>	<b>27</b>
6.1 Limitations	27
<b>References</b>	<b>29</b>
<b>Appendix</b>	<b>31</b>

# 1. Introduction

In today's globalized society, where an overwhelming variety of cuisines and dining options are available, making food choices can be a complex and time-consuming process. Consumers worldwide face an endless selection. However, finding the best option often requires significant effort. Various cultural and regional factors influence taste preferences and consumption behavior, making food selection even more complicated. For instance, individuals may choose between local authentic cuisine or international food options based on their interests and mood.

Practical factors such as budget, location, and restaurant operating hours also play a crucial role in food decision-making. Budget constraints can limit options for many people, as certain restaurants may be unaffordable [1]. Additionally, time constraints and differing preferences within a group can further complicate decision-making, particularly for families who must find a place that satisfies both children and adults while staying within budget and being geographically convenient [2].

Technological advancements have provided solutions to simplify food selection. However, many existing applications, such as Google Maps<sup>1</sup> and TripAdvisor<sup>2</sup>, offer general restaurant information without focusing on personalized recommendations. This is where the potential of artificial intelligence (AI) comes into play, enabling the consideration of almost all potential and effective factors to provide customized solutions to one's needs [3].

This thesis aims to design and evaluate a system, "AI-Powered Food Safari", that uses AI and Natural Language Processing to improve the food selection process for travellers. "**AI-Powered Food Safari**" is designed for modern travellers looking for a quick and high-quality dining solution. The system's core functionality lies in analyzing user preferences, budget, location, and time constraints to suggest the most suitable restaurants. It analyzes restaurant reviews, popularity, proximity, and operating hours. This system enhances user experience and reduces decision-making stress and time consumption, eliminating the need to browse long lists or read multiple reviews.

---

<sup>1</sup> [www.google.com/maps](http://www.google.com/maps)

<sup>2</sup> [www.tripadvisor.com](http://www.tripadvisor.com)

## **2. Background**

### **2.1 Natural Language Processing in Review Analysis**

Natural Language Processing (NLP) is one of the most significant applications in analyzing user-generated textual data, especially when it involves reviews. Reviews posted online contain a lot of rich, unstructured content about customer experience, preferences, and sentiments, which are basic for personal and context-sensitive recommendations [4].

In restaurant recommendations, NLP techniques help derive semantics from text, classify sentiments, and identify significant aspects of user feedback. In aspect-based sentiment analysis (ABSA), for example, systems associate specific aspects with their corresponding sentiments, such as food quality, service, ambiance, or pricing. It provides a deeper understanding than general sentiment classification, which is particularly reductive for user opinions [5].

Models that make advanced use of Transformer architectures, such as BERT (Bidirectional Encoder Representations from Transformers), RoBERTa, and GPT, are elevating the capabilities of systems in the comprehension of natural languages by capturing deep contextual relationships between words [6][7][8]. These models can be fine-tuned for specific NLP tasks such as sentiment classification, keyword extraction, and summarization, which are highly relevant for interpreting large volumes of restaurant reviews.

Besides this, using NLP techniques, a model can help decipher user prompts against natural language. For instance, a user queries: "Find a quiet Italian restaurant with good vegetarian options." When inputted, the NLP model will extract the intent: quiet, Italian, and vegetarian options, which correspond to the relevant review data, plus metadata [9].

Therefore, by applying such means, NLP can help make travel assistant systems transcend just keyword matching and provide personalized recommendations so closely aligned with the personalization of the user preferences, the emotional tone they carry, and contextual needs [10].

## 2.2 Data Mining

Data mining plays a vital role in developing intelligent recommendation systems driven by data by identifying patterns, relationships, and insights from structured data and unstructured sources of data. In the context of restaurant recommendation engines, data mining forms the basis of extracting relevant information from vast amounts of data encompassing user reviews, restaurant metadata, as well as contextual attributes like location or pricing [11].

One of the important phases in data mining is data acquisition, which can be approached via various methods depending on the data availability, format, and access policy. A traditional technique is web scraping, whereby the scripts would extract data directly from websites. This is particularly useful when sourcing information from platforms that do not have APIs or when some custom data selection is required, e.g., extracting selective types of reviews or certain attributes such as ambiance description or user-uploaded photos [12]. While web scraping is mighty and flexible, it requires incessant maintenance, as the templates for websites change frequently. Also, ethical and legal grounds must be considered since many websites actively restrict or discourage scraping in their terms of service [13].

An alternative to web scraping considerations is a stringently structured and controlled environment, e.g., the TripAdvisor API [14] or the Google Places API [15]. These give developers access to rich, structured, and well-maintained data sources, such as restaurant names, locations, ratings, reviews, and operational metadata. Furthermore, APIs shine in terms of stability and documentation, enabling developers to integrate them smoothly into their applications and perform data parsing without tedious consideration. However, they often impose restrictions, such as limited request levels, restricted granularity of data (especially at no charge), or limited access to complete review content, all creating bottlenecks for deep natural language analysis.

Another potential source of data involves the use of third-party tools and open datasets that provide pre-aggregated information useful for prototyping or supplementing core data pipelines. Tools such as Octoparse<sup>3</sup> and Apify<sup>4</sup> offer barely any coding for scraping workflow creation, while sites such as Kaggle<sup>5</sup> or academic repositories present datasets faster for experimentation and model training [16]. Although these sources offer a convenient starting

---

<sup>3</sup> [www.octoparse.com](http://www.octoparse.com)

<sup>4</sup> [www.apify.com](http://www.apify.com)

<sup>5</sup> [www.kaggle.com](http://www.kaggle.com)

point, they may not relate to a real-time or exactly fit specific user needs, requiring further refinement and preprocessing.

Such successful mining of data, within the scope of this instance, means that a mix of all the options maximizes coverage and data quality. For instance, metadata might be reliably obtained via API, while more delicate review content could be captured by scraping or data mining from open datasets [17]. Used together, these methods allow for a more comprehensive, representative dataset, leading to advanced analyses and more personalized, accurate restaurant recommendations.

## **2.3 Prompt Engineering**

### **2.3.1 Introduction to Prompt Engineering**

Prompt engineering is the practice of designing and optimizing input queries to guide large language models (LLMs) toward generating accurate, relevant, and task-specific responses [18]. As LLMs are highly sensitive to the phrasing and structure of their inputs, the way a prompt is written can significantly affect the output's quality. In AI systems that rely on natural language interaction, such as travel or restaurant recommendation assistants, prompt engineering is a key component in enabling meaningful conversations and reliable suggestions [3].

Unlike rule-based systems, LLMs generate responses based on context, examples, and implied intent. Prompt engineering is used to bridge the gap between the user's natural input and the structured output expected from the model. It helps disambiguate vague queries, inject context (such as location or preferences), and ensure consistency in recommendations. Especially in systems like AI travel assistants, it enables the model to personalize answers by incorporating metadata, previous interactions, or inferred user intent [20]. Without well-designed prompts, even powerful models may produce generic or inaccurate results [27].

### 2.3.2 Prompt Types and Techniques

Several prompting strategies are commonly used to guide LLM behavior [19][20]:

- **Zero-shot prompting:** Directly instructs the model without examples.  
*Example:* “Recommend a family-friendly Thai restaurant in Amsterdam.”
- **Few-shot prompting:** Provides examples in the prompt to guide output style and format.
- **Instruction-based prompting:** Gives explicit task directions.  
*Example:* “Summarize the top three positives and negatives from these reviews.”
- **Chain-of-thought prompting:** Encourages the model to explain its reasoning step-by-step before concluding. Useful for more complex decision-making.
- **Contextual prompting:** Includes metadata such as user preferences, location, time, or dining purpose to personalize responses.

These strategies are often combined to tailor LLM behavior to specific use cases. In this project, prompt engineering is applied to interpret user queries, enrich them with contextual data, and generate meaningful restaurant suggestions and review summaries. It is the foundation for aligning user intent with AI capabilities [3].

## 2.4 Related Works

The value of intelligent recommendation systems is demonstrated in several related projects dealing with the considerations of AI and NLP in aiding user decision-making about food and travel.

One such thesis is "*Päevapakkumiste soovitussüsteem juturoboti kujul*" [26], a chatbot developed to recommend daily lunch offers to employees through Slack. It acquires lunch offers data from outside APIs to adapt its recommendations based on historical patterns of individual users. Specific to food choices only in working hours, yet with excellent techniques such as incorporating communication channels, automated personalized recommendations, and addressing the cold-start issue for new users. Such facets may be very valuable references when it comes to developing travel assistance with AI, where personalization and platform integration are also vital concerns.

Another closely related development is the continuous work of Uber Technologies Inc.<sup>6</sup> on an AI chatbot that will be integrated into the Uber Eats app. It was found that this Uber AI assistant recommends relevant dishes in restaurants suited to the user's tastes, budget, and context [22]. This exemplifies the latest trend of embedding AI directly into popular consumer apps for more convenience and speed in decision-making. Also, this emphasizes the need for quick, easy interaction during high-frequency tasks like food ordering—an essential learning point for travel assistance, where timely and relevant AI-driven responses are also crucial.

Alongside these developments, food delivery apps like Bolt Food<sup>7</sup> and Wolt<sup>8</sup> have become popular across Europe for their user-friendly interfaces and access to various restaurants for delivering food to the customer's doorstep instead of enjoying the food at restaurants with users' tailored preferences. Hence, while these apps excel in logistics and availability, they often lack deeper personalization capabilities. For instance, users are generally presented with basic filters (e.g., cuisine, price, ratings) and, at the same time, receive a limited semantic understanding of their preferences, context (like mood or occasion), or dietary goals.

Therefore, the current recommendation systems remain predominantly static or based on some generalized popularity measures instead of more personalized sentiment analysis of user reviews or the understanding of individual behavior in its nuances. Hence, we foresee an opening for AI-based systems with the potential to utilize NLP, prompt engineering, and contextual data to issue real-time recommendations that appear more humanlike, flexible, and adaptive. Moreover, the limited geographic presence of such food delivery apps, particularly in small cities and rural areas, creates an opportunity for developing an app that does not heavily rely on a logistics setup but focuses only on the users' tailored preferences.

---

<sup>6</sup> [www.uber.com](http://www.uber.com)

<sup>7</sup> [www.food.bolt.eu](http://www.food.bolt.eu)

<sup>8</sup> [www.wolt.com](http://www.wolt.com)

### 3. Method and System Development

#### 3.1 Data Collection

The first step toward developing a dependable restaurant recommendation system is to gather a large and representative dataset of user reviews and metadata regarding restaurants in a specific geographic location as a case study. In our case, being a resident and the second largest city in Estonia, Tartu was chosen as the case study. Each restaurant on Google Maps has a unique identifier called a Place ID. Thus, collecting around 100 such IDs for local dining venues commenced. Apify was employed to automate this procedure, successfully identifying 101 eateries in Tartu.

As the next step, the Google Maps API<sup>9</sup> is a solid choice to collect user reviews; however, it was not fit for this task as it yields a maximum of five reviews per restaurant, which is too small a number for any meaningful sentiment analysis to take place. This limitation was avoided by using the Outscraper platform, which allows for the collection of many more reviews per venue, as indicated in Fig. 1. In total, around 20 reviews per restaurant were collected by using Outscraper, leading to approximately 2000 reviews in total, enough for carrying out a robust aspect-based sentiment analysis.

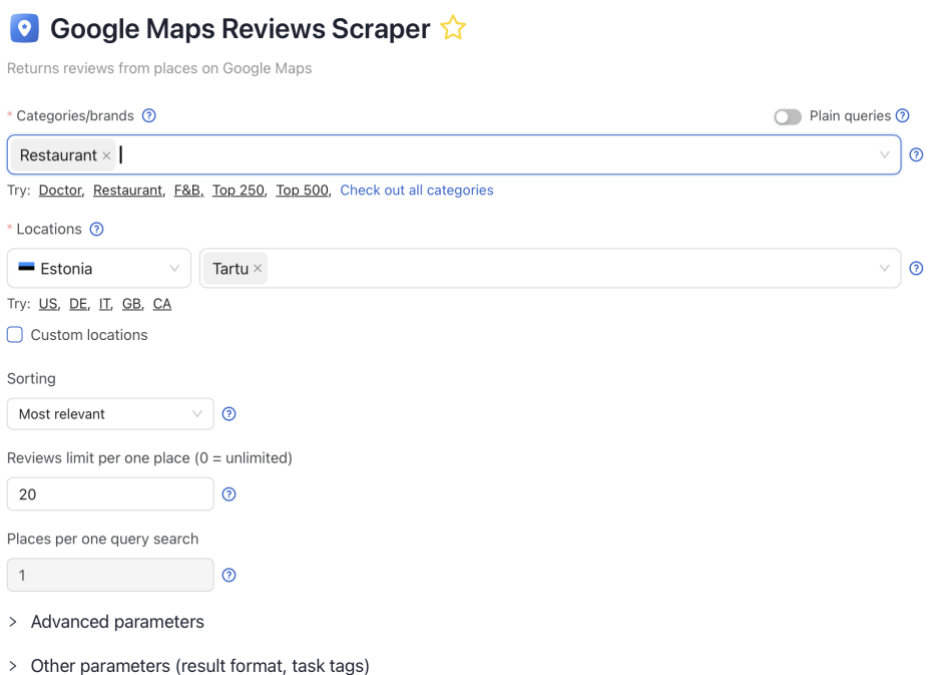


Figure 1. Outscraper template view.

<sup>9</sup> developers.google.com/maps

To avoid violations of user privacy rules, no user-sensitive data was collected.

Using the Google Places API, restaurant metadata was collected, which included average rating, price level, geographic coordinates, and opening time. After collecting the data, basic data cleaning was performed. This included removing empty or incomplete rows, extracting only the necessary columns such as review text and rating, and converting the cleaned datasets from CSV format into plain text files for easier processing in later stages of the pipeline. These steps ensured that the data was structured, relevant, and ready for sentiment and aspect analysis (see Table 1).

Table 1. Dataset statistics.

Parameter	Before data cleaning	After data cleaning
Number of restaurants	101	98
Number of reviews collected	1943	1881
Average number of reviews per restaurant	≈19	≈19

### 3.2 Evaluation: Use cases

To better understand the problem and the requirements for building a suggestion-based restaurant ranking system, the following use case examples were designed to reflect typical user scenarios and test core functionalities:

1. **Scenario:** The user wants a restaurant within a 3-mile radius.

**User Actions:**

- a. Opens the app and inputs: *"Show me good burger places within 3 miles."*

**System Actions:**

- b. Retrieves the current location.
- c. Filters burger places within 3 miles.
- d. Ranks based on ratings and user reviews.
- e. Displays the best options.

**Expected Outcome:** The user finds burger places close by with good ratings.

2. **Scenario:** The user only wants restaurants rated 4.5 stars and above.

**User Actions:**

- a. Inputs: *"Find me a sushi restaurant with at least 4.5 stars."*

**System Actions:**

- b. Fetches sushi restaurants.
- c. Filters those with 4.5+ star ratings.
- d. Sorts by highest reviews.
- e. Displays top options.

**Expected Outcome:** The user finds high-quality sushi places.

3. **Scenario:** The user wants affordable dining.

**User Actions:**

- a. Inputs: *"Find a good but affordable Mexican restaurant near me."*

**System Actions:**

- b. Retrieves nearby Mexican restaurants.
- c. Filters by price range (\$ or \$\$).
- d. Ranks based on ratings and affordability.

**Expected Outcome:** The user finds a good-value restaurant.

### 3.3 LLM-based Aspect Extraction

After the data collection process, we extracted the aspects mentioned in the collected reviews using an AI-human loop strategy, which has been shown to be effective in the literature [23]. To retrieve a significant number of relevant aspects, we employed LLMs—DeepSeek<sup>10</sup>, Gemini<sup>11</sup>, and Phi<sup>12</sup>—accessed via OpenRouter’s gateway. Each model was prompted using a specifically designed template (see Figure 2) to list all aspects discussed in the reviews, labeling each as either explicitly or implicitly mentioned. The outputs from the three models were collected as separate files and then combined, with DeepSeek employed again to consolidate and clean the list by removing duplicates or overly similar entries. To ensure comprehensive coverage across all relevant use cases, an additional prompt was used to supplement the aspect list based on predefined use cases. Finally, the authors manually verified and refined the consolidated list, removing any sparse, redundant, or irrelevant aspects. This

---

<sup>10</sup> [www.deepseek.com/en](http://www.deepseek.com/en)

<sup>11</sup> [gemini.google.com](http://gemini.google.com)

<sup>12</sup> [azure.microsoft.com/en-us/products/phi/?msocid=146532b29f536d5f314326b39edc6c53](http://azure.microsoft.com/en-us/products/phi/?msocid=146532b29f536d5f314326b39edc6c53)

human supervision step helped maintain the relevance and clarity of the dataset, resulting in a final list of 34 well-defined aspects (see examples in Table 2) to support the development of tailored solutions.

Table 2. Examples of aspects from the final aspects file.

Aspect	Category	Description
Romantic Ambiance	Implicit	The atmosphere is suited for couples and intimate settings.
Location	Explicit	The physical position of the restaurant and its convenience.
Late-Night Dining Suitability	Implicit	Openness during late hours and suitability for night dining.
Beverage Quality	Explicit	Quality and variety of drinks, both alcoholic and non-alcoholic.

```
# TASK OVERVIEW
Consider yourself as an expert on Natural Language Processing related issues.

# INPUT
- REVIEWS_FILE: {reviews_file}
- INPUT Format file: < > <TAB> < >
- INPUT LANGUAGE: Input text is in one of the languages or the mixer of this [English, Estonian, Russian]

# DESIRED OUTPUT FORMAT
1. ASPECT: "<aspect-name1>" Category: [implicit or explicit]
2. ASPECT: "<aspect-name2>" Category: [implicit or explicit]
so on ...

# INSTRUCTIONS
Taking into account the task overview and given reviews (i.e., REVIEWS_FILE), do the following steps:
1. Analyse and identify the important factors that are considered by users about a restaurant.
2. Propose crucial explicit as well as implicit aspects for finding the best restaurants.
3. Generate output strictly based on the DESIRED OUTPUT FORMAT.]
```

Figure 2. Aspect extraction template.

## 3.4 AI-assisted Recommendations

### 3.4.1 Ranking Model

The success of this thesis depends heavily on how correctly the ranking model assesses and combines user preferences in terms of aspects and restaurant metadata. To this aim, a ranking formula was proposed and refined several times to obtain the best solutions, leveraging a reasoning LLM and a subset of use cases. The open-source Deepseek R1 was employed as a reasoning LLM.

$$S(R_i) = W_1 \cdot avg\_ReviewScore + W_2 \cdot RestaurantMetadataScore$$

where,  $W_1 + W_2 = 1.0$

We leverage the reasoning LLM to propose weight values with justifications. After several executions, we considered the default values of weights:  $W_1 = 0.75$  and  $W_2 = 0.25$ .

Hence, the ranking score  $S(R_i)$  is a weighted combination of  $avg\_ReviewScore$  and  $RestaurantMetadataScore$ . However, the weights of  $W_1$  and  $W_2$  could be changed if the Ranking model finds that one or another score part should have an advantage.

***avg\_ReviewScore:*** The *avg\_ReviewScore* is computed first by performing aspect-based sentiment analysis on user reviews. The system refines aspects that have been predefined, analyzing every review in relation to these aspects with an attendant sentiment score assigned to each based on polarity (positive, neutral, or negative) and normalized to a scale of 0.1 to 1.0.

Next, the individual weights for each aspect are dynamically assigned according to the user's prompt. For example, if a user emphasizes "good service" or "quiet atmosphere", the associated aspects will take higher weights. Each resultant score is then multiplied by its respective weight, and the average score over all the reviews for that restaurant is calculated to form the final *avg\_ReviewScore*.

***RestaurantMetadataScore:*** The *RestaurantMetadataScore* is derived from the restaurant's metadata and is influenced directly by the user's prompt. Suppose the prompt includes constraints like "near me", "family-friendly", or "open late". In that case, the system prioritizes restaurants that satisfy those conditions, normalizing and combining the relevant metadata attributes into a score on the same scale.

By combining these two scores using the weighted formula, the model produces a ranked list of restaurants best aligned with the user's preferences.

### 3.4.2 Template Model Development

The core idea behind the model is to process the provided data files, analyze their contents, and assign a final score to each restaurant based on both customer sentiment and metadata (see Figure 3).

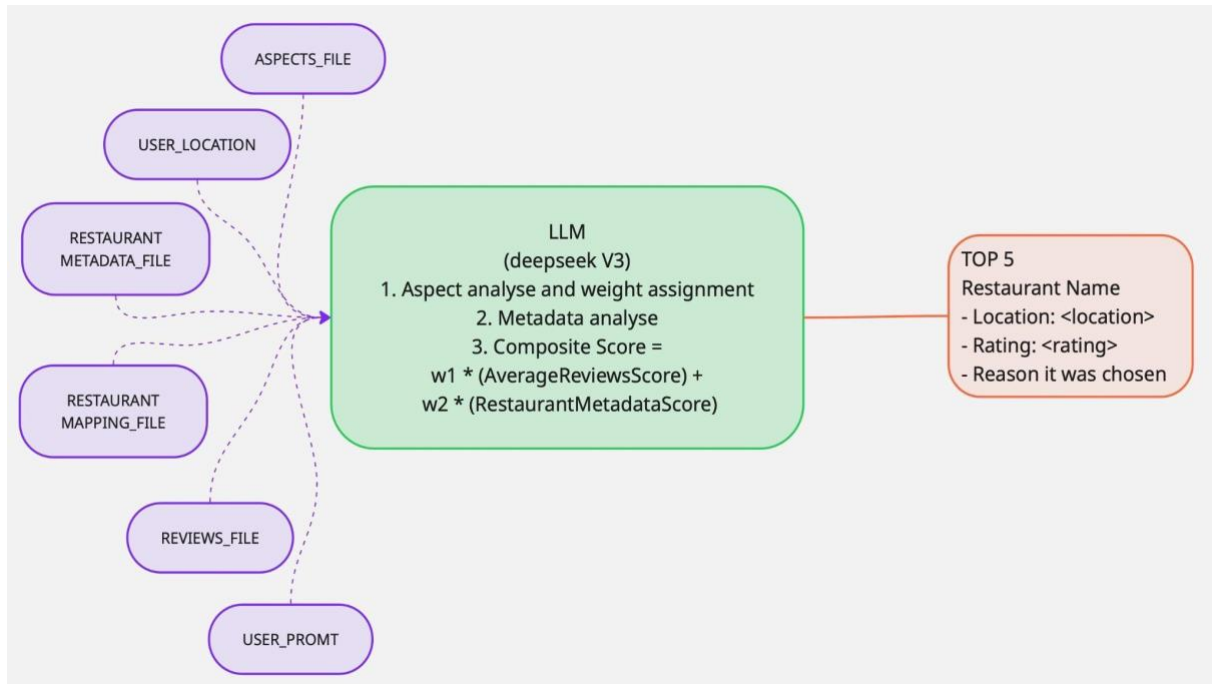


Figure 3. Main template model.

### 3.4.3 Prompt Template Description for Recommendation

The prompt template consists of 5 parts: Task Overview, Inputs, Desired Output Format, Ranking Formula, Detailed Instructions, and Additional Considerations.

#### Task Overview:

The task overview provides a summary of the template content and the assigned task. The objective is to produce a ranked list of top restaurants that best match the user's criteria, with careful consideration of sentiment and relevant metadata.

```
# TASK OVERVIEW

You are an NLP expert tasked with evaluating and ranking restaurants based on customer reviews, metadata, and user preferences. Your goal is to generate a ranked list of the best restaurants that align with the user's prompt. The ranking considers both sentiment from reviews and key metadata factors. Go step-by-step.
```

Figure 4. Task Overview.

## Inputs:

The input part consists of files and variables based on which the LLM performs its calculations. See the detailed overview in Table 3.

Table 3. Template input files.

Input	Format	Purpose
ASPECTS_FILE	ASPECT: "<aspect-name>" Category: [explicit or implicit]	Lists the restaurants' aspects
USER_LOCATION	<user_location>	Used for proximity-based calculations during ranking
REVIEWS_FILE	<review_id> <TAB> <review_text>	Restaurant reviews to analyse
USER_PROMPT	<user_prompt>	A natural language query specifying user preferences or requests, ex. "I would like to eat a pizza."
RESTAURANT_META_DATA_FILE	<restaurant_name> <TAB> <totalScore> <TAB> <location_lat> <TAB> <location_lng> ...	Pasta Palace <TAB> 4.7 <TAB> 40.7128 <TAB> -74.0060 <TAB>
RESTAURANT_MAPPING_FILE	<short_name> <TAB> <Full Name>	Stores mapping restaurant names from the Reviews file to shorten the context of the Reviews file.  Example: R1 Fusion Kebab R2 Kõverkiosk

## Desired Output Format

The final output of the restaurant ranking system is a curated list of the **Top 5 Restaurants**, each entry meticulously formatted to provide clear and concise information. For every restaurant, the output includes:

- **Restaurant Name:** The full name of the establishment, accurately mapped from the RESTAURANT\_MAPPING\_FILE.
- **Location:** Coordinates of a location, which can be used to recognize and assist in a user's journey, permit the identification of a location.
- **Rating:** An aggregated score reflecting the overall quality, derived from both sentiment analysis of reviews and relevant metadata.

- **The reason it was chosen:** A succinct explanation highlighting the key factors that contributed to the restaurant's ranking. This may encompass aspects such as high sentiment scores in critical areas (e.g., food quality, service), proximity to the user's location, availability during desired times, and alignment with specific user preferences indicated in the prompt.

This structured output ensures that users receive personalized and context-aware recommendations, simplifying their decision-making process by presenting the most relevant dining options based on comprehensive data analysis.

## Ranking Formula

It is possible to learn more about the ranking formula algorithm in [Chapter 3.4.1](#). Also see Figure 5.

```
# RANKING FORMULA

Calculate the composite score for each restaurant as follows:
| Composite Score = w1 * (avg_ReviewsScore) + w2 * (RestaurantMetadataScore)

- Default Weights:
| - w1 = 75% (Reviews and aspect sentiment analysis)
| - w2 = 25% (Restaurant metadata)
```

Figure 5. Ranking Formula.

## Detailed Instructions:

### 1. Parsing the Reviews File

The system initiates the process by scanning the reviews file. It filters out any entries that are empty or contain invalid data to ensure that only valid and relevant reviews are processed further.

### 2. Sentiment Analysis

The system analyzes the sentiment of each review, focusing on specific aspects like food quality, service, and ambiance. Each sentiment is classified (positive, neutral, or negative), translated into a numerical score ranging from 0.1 to 1.0, and weighted according to the user's preferences. It then calculates an average sentiment score per restaurant, ensuring the aspect weights sum to 1.

### 3. Calculating Restaurant Metadata Score

Several restaurant attributes are scored:

- **Proximity:** Restaurants closer to the user receive higher scores, especially if the user specifies "near me".
- **Availability:** Only restaurants open at the specified user time are considered.
- **Takeout/Delivery:** Higher scores are assigned to restaurants that offer these services when requested.
- **Restaurant Type:** Preferences for restaurant styles (e.g., fast food, family-friendly) are factored in.
- **User Ratings:** Both average star ratings and the number of reviews are incorporated.
- **Accessibility:** Restaurants with wheelchair access are scored higher when relevant to the user's request.

These scores are normalized and combined to produce a final metadata score for each restaurant.

### 4. Composite Score & Ranking

The system calculates a composite score for each restaurant by combining the sentiment and metadata scores. Restaurants are then ranked from highest to lowest based on this composite score.

### 5. Output Generation

The top 5 ranked restaurants are formatted and presented as the final output. Each listing includes a concise explanation of the factors that contributed most significantly to its ranking.

See Figure 6 for a more detailed overview of the template's Detailed Instructions part.

```

# DETAILED INSTRUCTIONS

1. Parse the REVIEWS_FILE:
  - Read each review.
  - Skip reviews that are empty or contain invalid data.

2. Sentiment Analysis:
  - For each review, identify sentiments for all aspects defined in the ASPECTS_FILE.
  - Evaluate the sentiment polarity (positive, neutral, negative) for each aspect.
  - Convert the sentiment into a numerical score (0.1 to 1.0).
  - Based on the user prompt, assign specific weights to each aspect (e.g., food quality, service) if indicated by the user's preferences.
    In total all aspects should have value weight value 1.
  - Compute the avg_ReviewsScore for each restaurant by averaging the sentiment scores multiplied by weights across all relevant reviews.

3. Calculate RestaurantMetadataScore:
  - Proximity:
    Normalize the distance between the user's location and the restaurant on a scale of 0 to 10 (closer locations receive a higher score).
    Adjust the weight for proximity if the prompt emphasizes "near me."
  - Availability:
    Verify that the restaurant is open at the user's specified time (e.g., Sunday evening). Exclude restaurants that are closed.
  - Takeout/Delivery:
    Increase the score for restaurants offering takeout/delivery if the user requests these options.
  - Restaurant Type:
    Consider user-specified preferences (e.g., family-friendly, fast food).
  - User Ratings:
    Factor in both the overall rating (1 to 5 stars) and the total number of reviews.
  - Accessibility:
    Give a higher score to restaurants that are wheelchair-accessible if this is mentioned in the prompt or metadata.
  - Normalize and combine these elements into the final RestaurantMetadataScore.

4. Composite Score & Ranking:
  - For each restaurant, compute the Composite Score using the provided formula.
  - Rank restaurants in descending order based on the Composite Score.

5. Generate Output:
  - Format the top 5 ranked restaurants in the desired output format.
  - Include a brief explanation for each restaurant, referencing the key factors (e.g., sentiment analysis, proximity, availability)
    that led to its ranking.

```

Figure 6. Detailed Instruction.

### Additional Considerations:

This section elaborates on some essential characteristics of the system, ensuring robustness and user relevance. It insists on the provision of multilingual reviews requiring accurate sentiment analysis for English, Estonian, and Russian. It also furthers states adaptability - prompt-specific preferences should influence the filtering and weighting of scores. Finally, it enforces extreme data quality by disallowing invalid reviews and closed restaurants at the time of the user query (see Figure 7).

```

# ADDITIONAL CONSIDERATIONS

- Multilingual Reviews:
  Ensure that sentiment analysis correctly handles reviews in English, Estonian, and Russian.
- User Prompt Adjustments:
  Modify weights or filters (such as prioritizing proximity) if explicitly mentioned in the user prompt.
- Exclusions:
  - Do not consider reviews that are empty or invalid.
  - Exclude restaurants that are closed at the time specified by the user.

```

Figure 7. Additional Considerations.

## 4. Evaluation and Results

### 4.1 Evaluation Setup

To assess the performance of the AI-based restaurant recommendation system, a user-centered evaluation was conducted using 14 distinct use cases (Please see Table 4). Each use case represents a unique user profile or scenario and was tested by one real user. The goal was to determine whether the system could generate relevant restaurant recommendations tailored to the user's preferences.

Each user was presented with a Top-5 list of recommended restaurants, generated by the system based on their input preferences (e.g., cuisine, budget, location, etc.). The user then evaluated the recommendations and identified whether a suitable restaurant—the one they would realistically consider or prefer—was included in the Top-5 list.

Table 4. Summary of the use cases.

Case No.	Input	Expected Output
1	“Find the best Italian restaurant near me.”	The user finds Italian restaurants with high ratings.
2	“Show me good burger places within 100m.”	The user finds burger places close by with good ratings.
3	“Find me a sushi restaurant with at least 4.5 stars.”	The user finds high-quality sushi places.
4	“Suggest a restaurant for dinner.”	The user finds high-rated restaurant which is open in the evenings and serves great dinner
5	“Find a good but affordable Mexican restaurant near me.”	The user finds a good-value Mexican restaurant.
6	“Find a late-night restaurant near me.”	The user finds dining options that are available after 22.00
7	“Find me a cozy indoor café near me.”	The user finds an indoor café for a comfortable meal.
8	“Find a romantic restaurant for an anniversary dinner.”	The user finds a perfect place for a romantic evening.
9	“Find a restaurant with outdoor seating.”	The user finds a scenic outdoor dining experience.

10	“Find a kid-friendly restaurant nearby.”	The user finds a child-friendly place for dining.
11	“Find a restaurant that can accommodate 10 people.”	The user finds a place for a large group dinner.
12	“Find restaurants with live music near me.”	The user finds a lively dining spot.
13	“Three students want to go for lunch at noon.”	The user finds an affordable place with great lunch deals
14	“I would like to have a family dinner at 6 pm.”	A user finds a restaurant that is open after 6 pm., and the review mentions it is family-friendly

## 4.2 Evaluation Metric: Top-5 Accuracy

The primary evaluation metric used was Top-5 Accuracy, a standard metric in recommendation systems. It measures the proportion of times the user's correct or preferred restaurant appeared among the system's top 5 recommendations.

The metric is computed as follows:

$$\text{Top-5 Accuracy} = \frac{\text{Number of Successful Recommendations}}{\text{Total Number of Use Cases}}$$

A recommendation is considered successful if the user's selected or preferred restaurant is found within the Top-5 list generated by the system.

## 4.3 Results

The evaluation consisted of 14 individual test cases, with each use case representing a distinct user scenario. For each case, the system generated a ranked list of the Top-5 restaurant recommendations based on the user's input preferences.

After reviewing the recommended list, each user was asked to select whether at least one of the suggested restaurants matched their actual preference, expectation, or intention. This evaluation served to simulate a real-world decision-making process, where users are presented with multiple options and must determine if any of them are viable choices.

Breakdown of Results: 13 out of 14 users ( $\approx 92.9\%$ ) confirmed that at least one of the recommended restaurants aligned with their choice or preference. These are considered "hits".

1 out of 14 users ( $\approx 7.1\%$ ) did not find their preferred restaurant or a sufficiently relevant alternative in the Top-5 list. These are considered "misses".

**Interpretation:** The obtained results indicate a high level of effectiveness for the proposed ranking approach for recommendation. The high Top-5 accuracy reflects strong practical utility, as users typically consider only the first few recommendations when making a selection. The high hit rate also suggests that most users are likely to be satisfied with the recommendations and could quickly find a restaurant they like, improving overall user experience. Nonetheless, the presence of a single miss highlights the potential for further refinement, particularly in accommodating outlier preferences or enhancing personalization for edge cases.

#### 4.4 Demonstration of 2 Use Cases:

To demonstrate the practical effectiveness of the system, I'd like to include a few output examples. These examples showcase how user prompts are interpreted and how the system generates a personalized list of top restaurant recommendations based on sentiment analysis and metadata.

Example 1:

Prompt: "Three students want to go for lunch at noon."

Output:

##### 1. Restoran Hölml

- Location: 58.3795318, 26.7217236
- Rating: 4.8
- Reason: High ratings for food quality and service, excellent reviews mentioning delicious and innovative dishes. Open at 12:00, suitable for students.

##### 2. La Dolce Vita

- Location: 58.3818407, 26.7227316
- Rating: 4.6
- Reason: Authentic Italian cuisine with great pizza and pasta options. Positive reviews highlight friendly service and good value. Open at 12:00.

### 3. Kolm Tilli

- Location: 58.3703859, 26.7165481
- Rating: 4.5
- Reason: It is popular for its diverse menu and student-friendly atmosphere. Reviews praise the tasty food and quick service. It opens at 12:00.

### 4. Vilde Ja Vine

- Location: 58.3781198, 26.723041
- Rating: 4.6
- Reason: Great for groups, with a cozy atmosphere and good food. Reviews highlight generous portions and friendly staff. Open at 12:00.

### 5. Pompei

- Location: 58.380725, 26.7222505
- Rating: 4.6
- Reason: Excellent Italian-inspired dishes with fresh ingredients. Reviews mention creative flavors and good service. Open at 12:00.

Example 2:

Prompt: "I would like to have a family dinner at 6 pm."

Output:

#### 1. Restoran Spargel

- Location: 58.3746643, 26.7286911
- Rating: 4.7
- Reason it was chosen: Excellent for family dining with high-quality food and service. Reviews highlight its welcoming atmosphere, delicious dishes, and family-friendly environment. The restaurant is open at 6 PM and has a strong overall rating.

#### 2. Chez Andre

- Location: 58.380884, 26.7214445
- Rating: 4.6
- Reason it was chosen: Known for its cozy atmosphere and excellent food, Chez Andre is a great choice for families. Reviews mention its tasty meals and good service. The restaurant is open at 6 PM and has a high rating.

### **3. Hansa Tall & Hansa Hoov**

- Location: 58.3707297, 26.7390966
- Rating: 4.6
- Reason it was chosen: Offers a rustic and family-friendly dining experience with good food. Reviews highlight its generous portions and pleasant atmosphere. The restaurant is open at 6 PM and has a solid rating.

### **4. Pompei**

- Location: 58.380725, 26.7222505
- Rating: 4.6
- Reason it was chosen: An excellent option for families with its varied menu and good service. Reviews praise its delicious food and welcoming environment. The restaurant is open at 6 PM and has a high rating.

### **5. La Dolce Vita**

- Location: 58.3818407, 26.7227316
- Rating: 4.6
- Reason it was chosen: Perfect for family dinners with its authentic Italian cuisine and friendly service. Reviews highlight its tasty pizzas and pastas. The restaurant is open at 6 PM and has a strong overall rating.

## 5. Implementation and Further Developments

The restaurant recommendation system developed in this thesis has strong potential for real-world implementation as a mobile application or a web-based platform. Each format presents unique advantages, although both aim to make intelligent, personalized restaurant suggestions easily accessible to users based on their preferences, current location, and real-time needs.

### 5.1 Mobile Application

The **mobile application** can be very useful to a traveler or resident in need of recommendations. The application could also use the GPS feature in the device to dynamically configure search results with respect to the user's proximity to various restaurants. Users would enter their natural language and get results suitably ranked in accordance with their semantic analysis and metadata. With a clean, intuitive interface, the app would offer rich visual feedback, including maps, photos, and review summaries, making it a practical tool for daily use.

### 5.2 Web Application

The **browsing application** is complementary to the mobile version because it offers features that are more suitable for some people who want to specify restaurant information or plan restaurants before going out. The web interface may integrate applications such as Google Maps, booking systems for restaurants, or delivery providers. Moreover, through this web application, users could register accounts, save preferences, and track past queries. It is ideal for the most part for individuals constantly looking for specific restaurants with certain features or diet preferences. Multilingual support may also be added in line with the capability of the system to analyze reviews in languages other than the local language.

## **6. Conclusion**

This thesis explored the development of an intelligent restaurant recommendation system based on advanced artificial intelligence, more precisely, using natural language processing mechanisms along with aspect-based sentiment analysis to deliver personalized and context-aware recommendations. Combining different facets from user-generated reviews with structured restaurant metadata and flexible natural language prompts, the system aims to go beyond traditional approaches of offering location-based listings into one that derives more meaning and richness from the recommendations that consider the user's intent.

The heart of the solution is targeted at extracting sentiments across a range of explicit and implicit aspects from food quality to service speed, ambiance, and so on by prompt-engineering large language models. The prompt sentiment is weighted by the user's prompt reflecting their preferences, such that all final rankings represent every user's priority. The metadata source consists of information about the establishment: its location, ratings, opening hours, and so forth; these refine the scoring mechanism to yield a composite evaluation grounded in opinion melding with objective information.

In conclusion, the project shows how everyday decision-making can be enhanced using contemporary AI techniques. It provides a flexible, language-aware, and user-centric approach to restaurant discovery that bridges data and human experience.

### **6.1 Limitations**

An intelligent restaurant ranking system drawing upon NLP and structured metadata for the client, providing customized options for top-ranking restaurants based on the user query. Although this entire process seems reasonable and uses powerful large language models for aspect-based sentiment analysis and their corresponding metadata evaluation, some drawbacks affect the accuracy of the system, its scalability, and its performance.

#### **Google API Limitation**

One of the most important limitations here is that review data available through the Google Places API is limited to the five reviews most relevant for each restaurant. Consequently, this implies that sentiment analysis, which itself contributes 75% towards the final composite score, will be conducted on an extremely limited and possibly unrepresentative sample of customer

reviews. Given the fact that most restaurants have hundreds, if not thousands, of reviews, how could a snapshot of five be representative of the consumer experience for such nuanced or implicit terms as "ambiance" or "family-friendliness"? As a result, such scarce information on sentiment scores may result in bias or lower ranking accuracy.

### **Context Size Limitation**

Another technical constraint lies in the token and context length limits of large language models used in this process. When processing multilingual reviews across multiple aspects, context size becomes a bottleneck. To address this, the implementation introduces a restaurant mapping file that replaces full restaurant names with short codes (e.g., R1, R2) in the reviews. This allows more content to be packed into the model's context window during analysis. However, it introduces complexity in the preprocessing and postprocessing steps. After analysis, these codes must be mapped back to full names for human-readable output, increasing the risk of data mismatches or interpretation errors if not handled carefully.

### **Model Time Consumption**

Another constraint is processing time. Each review must be evaluated on as many as 34 aspects, with sentiment classified and rated from 0.1 to 1.0. This fine-grained, multi-aspect sentiment analysis weighs heavily in computation, especially when done across hundreds of restaurants across several reviews. Moreover, the review data is multilingual—English, Estonian, or Russian—which calls for the model to be cross-linguistic and contextually aware; this can slow down execution further and occasionally introduce the ambiguity of translation in sentiment interpretation.

### **Data Mutation**

The RestaurantMetadataScore, although easier to calculate, is limited because of the freshness and completeness of metadata. Whenever metadata is generated, opening hours, takeout options, or price levels are subject to change. However, unless the data source is constantly refreshed, the model may rate a restaurant high based on outdated or wrong information. For example, a restaurant may show as open on Sunday evenings in the metadata but has changed its schedule. Similarly, a restaurant listed as "wheelchair accessible" in the metadata may not be accessible to a proper degree, misinforming the user dependent upon such metadata.

## References

- [1] Blake Eligh (2017). *What's for dinner? New study reveals how low-income families choose what's on their plate*. U of T News. University of Toronto. 07.11.24.
- [2] Klaus G. Grunert (2002). *Current issues in the understanding of consumer food choice*. ScienceDirect. Trends in Food Science & Technology. 07.11.24.
- [3] Raj Rupesh Bharadwaj (2024). *AI-Powered Recommendation Systems: Enhancing Upselling Opportunities in Restaurants*. LinkedIn. 07.11.24.
- [4] Bing Liu (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers. 03.04.25.
- [5] Yan Cathy Hua, Paul Denny, Jörg Wicker & Katerina Taskova (2024). *A systematic review of aspect-based sentiment analysis: domains, methods, and trends*. 08.04.25.
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Association for Computational Linguistics. 08.04.25.
- [7] Yinhan Liu, Myle Ott, Naman Goyal, et al. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. Chinese Information Processing Society of China. 08.04.25.
- [8] Tom Brown, Benjamin Mann, Nick Ryder, et al. (2020). *Language Models are Few-Shot Learners*. NIPS Papers. 08.05.25.
- [9] Dan Jurafsky & James H. Martin (2021). *Speech and Language Processing (3rd ed., draft)*. Stanford University. 07.11.24.
- [10] Yongfeng Zhang, Huifeng Cheng, and Sheng Wang (2020). *Explainable Recommendation via Multi-Task Learning in Opinionated Text Data*. ACM TOIS. 07.11.24.
- [11] Jiawei Han, Micheline Kamber, and Jian Pei (2011). *Data Mining: Concepts and Techniques (3rd ed.)*. ScienceDirect. 07.11.24.
- [12] Matthew Russell (2018). *Mining the Social Web (3rd ed.)*. O'Reilly Media. 07.11.24.

- [13] Ryan Mitchell (2018). *Web Scraping with Python: Collecting Data from the Modern Web* (2nd ed.). O'Reilly Media. 07.11.24.
- [14] TripAdvisor API Documentation (2024). *TripAdvisor Developer Portal*. TripAdvisor. 07.11.24.
- [15] Google Places API Documentation (2024). *Google Cloud Platform*. Google. 07.11.24.
- [16] Ansell Barrett (2025). *10 Best Open Source Web Scrapers in 2025*. Octoparse. 07.11.24.
- [17] Kaggle (2024). *Restaurant Review Datasets for NLP*. Kaggle.com. 07.11.24.
- [18] Lilian Weng (2023). *Prompt Engineering Guide*. Lil'Log. 07.11.24.
- [19] Aishwarya Naresh Reganti (2024). *Applied LLMs Mastery 2024*. Notion. 07.11.24.
- [20] OpenAI (2023). *Best Practices for Prompt Engineering with GPT Models*. OpenAI Documentation. 07.11.24.
- [21] McKenzie, S. (2023). *From Prompt to Output: Optimizing GPT Interactions for Domain-Specific Applications*. Medium. 05.05.25.
- [22] Gurman, M., & Lung, N. (2023). *Uber Eats' New AI Chatbot Will Offer Recommendations to Customers*. Bloomberg. 05.11.24.
- [23] Amershi, S., Weld, D., Vorvoreanu, M., et al. (2019). *Guidelines for Human-AI Interaction*. Microsoft. 05.05.25.
- [24] Dagang Wei (2025). *OpenRouter: A unified interface for LLMs*. Medium. 05.05.25.
- [25] Wang, X., Liu, Z., & Yu, D. (2024). *DeepSeek and Gemini: Scalable Architectures for Cross-Domain NLP*. arXiv. 05.05.25.
- [26] Verner Läll. (2020). *Päevapakkumiste soovitusüsteem juturoboti kujul*. Graduation Theses Registry. 06.05.2025.
- [27] Patric. (2025). *Mastering System Prompts for AI Agents*. Medium 08.05.25.

## Appendix

### 1. Licence

#### **Non-exclusive licence to reproduce thesis and make thesis public**

I, **Anton Berik**,

(author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

**AI-POWERED FOOD SAFARI: RESTAURANT PICKS FOR THE MODERN TRAVELER,**

(title of thesis)

supervised by Somnath Banerjee.

(supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Anton Berik

*14/05/2025*