National College of Ireland

Emotion-Aware AI Meal Recommendation with Calorie-Constrained Recipe Scoring and Dual-Stage Deep Learning Models

MSc Research Project
MSc Artificial Intelligence

MOHAMMED NIHAL THONIKKARA
Student ID: 23350750

School of Computing
National College of Ireland

Supervisor: Abdul Razzaq

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	MOHAMMED NIHAL THONIKKARA		
Student ID:	23350750		
Programme:	MSc Artificial Intelligence	**Year:**	2025
Module:	Practicum Part 2		
Supervisor:	Abdul Razzaq		
Submission Due Date:	11-12-2025		
Project Title:	Emotion-Aware AI Meal Recommendation with Calorie-Constrained Recipe Scoring and Dual-Stage Deep Learning Models		
Word Count:	8596 **Page Count**22		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Nihal

Date: 11-12-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Emotion-Aware AI Meal Recommendation with Calorie-Constrained Recipe Scoring and Dual-Stage Deep Learning Models

MOHAMMED NIHAL THONIKKARA
23350750

Abstract

In fact, the contemporary world has grown used to the existence of smart solutions that can help justify everyday choices such as the food options that are determined due to the mood, the health objectives, and the convenience. This study project offers a simplified yet effective AI-based meal recommendation software that combines emotional insights and nutritional restrictions. The essence of the innovation is that it has a two-stage deep learning pipeline. The initial phase proposes a high-precision Bi-Directional Long Short-Term Memory (Bi-LSTM) model to classify the emotion based on the text of the user with a test accuracy up to 92.87. This emotional inference is an exquisite feature engineering technique, which allows the system to encode affective cues in the downstream decision-making. The second phase introduces a lightweight regression-based model that produces a preference score of every recipe on combining the inferred emotive status with the energy needs. This model achieves a Mean Absolute Error (MAE) of 0.189, which proves that this model is effective in enabling the joint effects of emotional context and nutritional needs to determine the perceived suitability of recipes. In spite of the fact that the current system is more based on emotion-oriented personalisation and calorie matching, the methodology offers a basis of future extensions, such as ingredient replacement, deeper contextual perception and multimodal input. The obtained insights provide empirical support on the potential viability of emotionally aware meal recommendation and a solid foundation of the intelligent food-planning assistant of the next generation.

1 Introduction

There is food to be eaten every day; and that simple need sparks up much more complicated decision-making than might at first seem. People and families make daily decisions influenced by mood, nutrition objectives, time and degrees of cooking confidence. This has been trying to be alleviated within the past few years using digital platforms and recommender systems. However, even with advancement, the majority of current solutions are generic, which means they provide their recommendation that is seldom tailored to the specifics of the situation as it presents itself to the user. Their inability to identify and use the emotional state of a user is a specific weakness because it is a factor that has a strong impact on food preferences and selection of the meals (Dey and Arefin, 2024).

Lack of emotional sensitivity in traditional systems tends to result in recommendations that are not in tune with one's current mood, lack of sensitivity and hence fail to offer personalised support in the true sense of the word. On the same note, suggestions, which are independent of nutritional needs, like calorie limits, might not be sufficient to make healthier or more appropriate meal choices. These loopholes provide plenty of space to intelligent systems with the ability to know users in a more holistic manner. To overcome these constraints, the current study will suggest a narrow but advanced AI-based meal-recommendation system merging

deep learning methods and affective computing. The system does not just provide recipes but hopes to decode the emotional expression of the user in an untyped natural language text and combine it with calorie needs to provide the contextually appropriate meal suggestions. The project focuses on two phases of machine-learning pipeline: an emotion classifier, which understands the user sentiment based on text, and a calorie-sensitive predictive component, which assesses and ranks recipes based on the projected emotional state.

The central research questions driving this project are the following:

1. To what extent can an emotion classifier implemented with deep learning understand meaningful emotional states of users with regard to personalised meal recommendation through user-generated text?
2. How do emotional and calorie-based specifications into a recommendation model enhance the correspondingness and relatability in recommended recipes?
3. What do the assessment of a two stage emotional-context pipeline reveal, and what would such a course of action tell us about the future of affect-aware recommendation systems?

1.1 Research Contribution

The primary contribution of this work is the design and empirically validated focused two-stage recommendation pipeline incorporating affective computing and nutritional awareness. During the first step, a model of emotion classification is derived based on quality textual data, thus allowing the system to judge the emotional state of a user with a lot of precision. This deduced emotion score can be viewed as a valuable contextual input, as a feature generator in the next recommendation module. This emotional representation is then joined with calorie data in the second stage to assist a lightweight form of regression-based model to provide a preference score to each recipe. The proposed pipeline contrasts with the traditional systems that use only textual or nutritional features to determine the sentiment; however, the concept of AI-inferred sentiment as a powerful and complementary predictor of increased relevance of meal recommendations proves to be useful. Interaction of nutritional and emotional indicators.

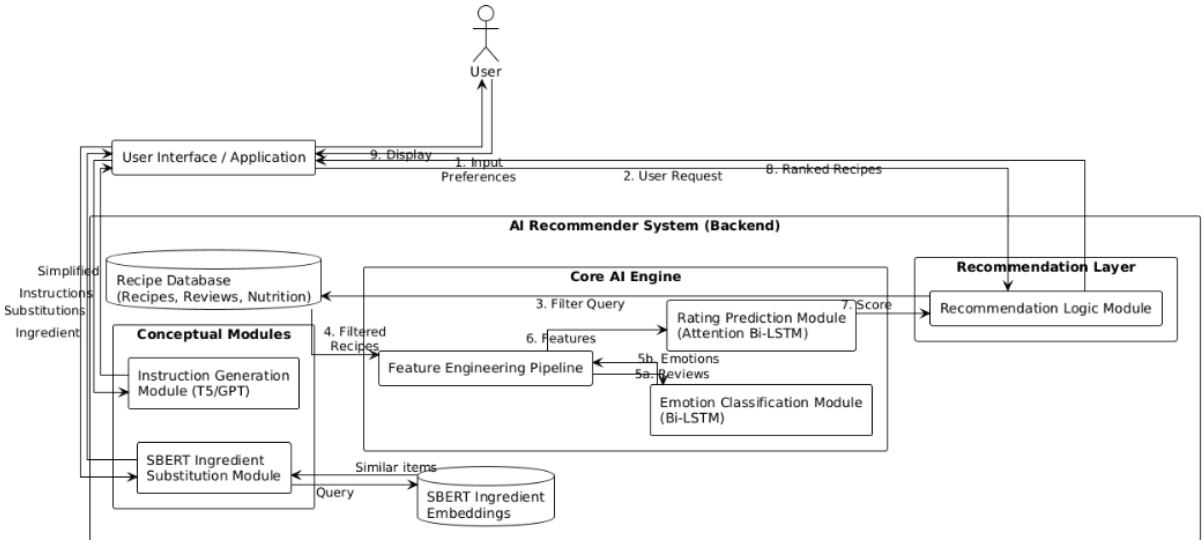

Figure 1: High-level architecture of the proposed multi-component AI meal recommender system.

1.2 Report Structure

The rest of this report is organized in the following manner. A survey of relevant literature in recommendation systems, ingredient substitution, and natural language processing within the culinary domain will be provided in Section 2. Section 3 discusses the research methodology with a description of the quantitative and experimental procedure adopted. Section 4 is dedicated to design specifications, providing details on the system architecture and the design of machine learning models. Section 5 discusses the implementation, outlining the building of a valid prototype in Python using relevant libraries. Section 6 provides an in-depth evaluation of the implemented models, assessing their performance according to different metrics. Finally, Section 7 wraps with the conclusions of the report, summarizing the findings, discussing limitations, and pointing important future work directions.

2 Related Work

The current study places itself at that larger context of scholarship and technology. It aims to present a backdrop for a critical analysis that will be subsequently presented under the following headings, which represent the major topical areas related with the proposed system: food recommender systems, intelligent ingredient substitution, and instruction analysis and generation through natural language processing. The current work brings a critical discussion of existing works and points out limitations that this project intends to fill.

2.1 Food Recommender Systems

The use of food recommender systems has changed over time-some early systems implemented simple filtering methods, whereas others had automated recommendations based on sophisticated machine learning-driven platforms. Most early systems relied either on collaborative filtering, in which items are recommended based on the preferences shown by users with similar profiles, or content-based filtering, which recommends items based on intrinsic attributes (Maftei, 2025). Although collaborative filtering is effective, it cannot make a recommendation for a new user or new items that have not been interacted with; this is referred to as the cold start problem. While content-based systems can solve these problems, their performance is often limited significantly due to the available metadata quality and scope. Newer research alludes to hybrid models that amalgamate both techniques. RecipeBERT brings personalized recommendations on food using BERT, demonstrating the extent to which deep learning-based language models can comprehend recipe texts. Their research aims at matching user preferences against the recipe descriptions. Likewise, Wang (2025) offers a general survey of Large Language Models (LLMs) in use for building ultimately more conversational and context-aware recommender systems. They understand a naturally spoken user query but do not explicitly model less tangible factors-including perhaps the quality of the recipe itself or the emotional state a meal may induce within it.

Novelty in recommendation is based on the use of additional data. Sha et al. (2025) used LLMs for diet recommendations according to Traditional Chinese Medicine principles and added a complex knowledge base and user health data. It marks a trend to more holistic and specialized recommendation engines. Following in this direction, the present research introduces new forms of auxiliary information: emotional sentiment extracted from user reviews. System

newness relies directly on that: different from systems solely based on star ratings toward more nuanced detection of user satisfaction signals, this is touchingly capable of predicting recipe quality.

2.2 SBERT-Based Ingredient Substitution

The significant issue in enhancement of recipe accessibility and adaptivity is that of ingredient substitution. For example, a person can't find a specific ingredient, and this should not pose a barrier to cooking. While considerable research has focused on this theme, a wide range of different techniques have been explored. Initial techniques largely relied on manually curated ontologies and knowledge graphs. This method aims at food substitution by using a knowledge graph of food, and thus offers high precision but suffers from the usual scalability problems and requires huge efforts to build and maintain the graph (Shirai et al., 2021).

At the same time, the introduction of deep learning has stimulated the interest for embeddingbased approaches, or representing each ingredient as a high-dimensional vector, wherein their proximity indicates semantic similarity. Morales-Garzón et al. (2022) used contextual sentence embeddings in order to retrieve the various versions of the same recipe, which indirectly pinpoints potential replacements. Unfortunately, traditional word embeddings such as Word2Vec or GloVe often don't capture the context in which an ingredient is used. For example, "milk" in a bakery recipe has a different function than "milk" in a savory sauce.

Such contextual limitations are erased by models such as SBERT and other transformer-based models through the generation of contextually aware embeddings. The authors Senath et al. (2025) specifically examined the application of LLMs in ingredient substitution through fine-tuning and preference optimization to generate suitable alternatives. The results confirm the superiority of modern language models for this task. Kim, Venkataramanan, and Sheth (2024) give an extensive review on the subject of food ingredient substitution, mapping methods and highlighting challenges. This methodology proposed through the present project is consistent with those cutting-edge ones by choosing SBERT as the core technology for understanding the functional role of an ingredient in a recipe context, thus ensuring more accurate and sensible culinarily relevant suggestions than models that are context agnostic. A comparative example regarding types of ingredients is provided in Table 1.

Table 1: General Comparison of Common Ingredient Categories and Their Functions

Category	**Primary Function(s)**	**Example Ingredients**	**Potential Substitutes**
Acids	Brightening, Balancing Flavor, Tenderizing	Lemon Juice, Vinegar, Buttermilk	Lime Juice, Wine, Yogurt
Fats	Flavoring, Moisture, Texture, Heat Transfer	Butter, Olive Oil, Lard	Margarine, Coconut Oil, Ghee
Leavening Agents	Lift and Airiness	Baking Soda, Baking Powder, Yeast	Egg Whites, Self-Raising Flour
Sweeteners	Sweetness, Browning, Moisture Retention	Granulated Sugar, Honey, Maple Syrup	Agave Nectar, Brown Sugar

Binders	Hold Ingredients Together, Provide Structure	Eggs, Flour, Breadcrumbs	Flax Egg, Cornstarch, Oats

2.3 Instruction Generation and Simplification

A final key feature of the system proposed is instruction generation or modification. Instruction clarity and simplicity are essential for successful cooking experiences for novices. Traditional recipe platforms provide a static set of steps of equal difficulty that do not change with the user's own abilities. Natural Language Generation corresponds to the generation of more dynamic, individualized instructions.

Yuan et al. (2023) have investigated distilling /script knowledge/ from LLMs for language planning, which consists of the understanding and generation of sequences of actions and is very much in line with the generation of recipe steps. Their findings suggest that LLMs have procedural knowledge internalized and can regenerate it in a structured form. The work by Veerabadran et al. (2025), which sets a benchmark for multimodal goal inference for assistive agents, with a special focus on egocentric video, share the same goal of breaking down complex tasks into manageable sub-tasks for the user, thus while not directly about recipes, the principles behind task decomposition are quite relevant in this case.

A generative mode might be induced in applying the ideas to the culinary setting by, e.g., rewriting those instruction for the beginner, possibly involving some definition of more technical terms (e.g., explaining what sauté means), splitting multi-part steps into smaller, singular actions, and giving explicit guidance on timings and visual cues. This project is envisioning at a conceptual level, the use of a pre-trained generative model such as T5 or GPT for this purpose, using the enormous amounts of world knowledge and generative processes these systems are capable of for changing standardized recipe instructions to guidance suitable for the user.

Thus, in summary, the existing literature portrays significant progress in the different areas of food recommendation, ingredient substitution, and natural language generation. Nonetheless, a big gap is found in the integration of these components into one framework. The other contribution, which is novel, is the one using AI-inferred emotion from user reviews as an important independent variable for prediction of recipe quality. The present work tries to address this gap by building a technically sound system that is holistically designed to address the real-life challenges of everyday cooking.

Table 2: Critical Review of Significant Related Work

Reference	**Focus / Topic**	**Methodology**	**Key Contribution(s)**	**Limitations / Gap Addressed by This Project**
Mereddy & Beedareddy (2024)	Personalized food recommendations for smart homes.	BERT-based model (RecipeBERT) for semantic understanding and matching	Showcased the effectiveness of transformer models in capturing the nuances of recipe	The system relies exclusively on textual data. It does not integrate multi-modal inputs like

		of recipes to user queries.	text for content-based filtering.	nutritional information or latent signals such as the emotional sentiment of user reviews. This project incorporates both.
Senath et al. (2025)	Ingredient substitution in food recipes.	Fine-tuned Large Language Models (LLMs) using supervised learning and direct preference optimization.	Demonstrated state-of-the-art performance in generating contextually appropriate ingredient substitutes, surpassing traditional embedding methods.	Focuses solely on the substitution task. The work does not integrate this capability within a broader recommendation system. This project proposes a methodology to embed this feature into a complete system.
Sha et al. (2025)	Dietary recommendations based on Traditional Chinese Medicine (TCM).	Retrieval-Augmented Generation (RAG) with LLMs, incorporating a specialized knowledge base (TCM).	A novel application of LLMs to a highly specialized domain, combining generative capabilities with domain-specific knowledge for health-oriented recommendations.	The approach is tailored to a niche and complex domain. The current project aims for a more universally applicable recommender system based on a more general signal (emotion).
Shirai et al. (2021)	Ingredient substitution using knowledge graphs.	Construction and querying of a food-centric knowledge graph to identify relationships	A structured, high-precision approach to substitution that leverages explicit, curated relationships between food items.	Knowledge graphs are difficult and labor-intensive to build and maintain, limiting scalability. This

		and potential substitutes.		project proposes using SBERT, which learns relationships from data and is more scalable.
Morales-Garzón et al. (2022)	Finding different versions or variations of food recipes.	Contextual sentence embeddings to measure semantic similarity between entire recipes.	An effective method for recipe clustering and versioning, which can indirectly suggest ingredient substitutions.	The method is not designed for direct, on-the-fly substitution of a single ingredient but rather for finding whole-recipe alternatives. The proposed SBERT module is designed for granular substitution.
Yuan et al. (2023)	Generating structured plans or scripts from LLMs.	A technique called "script knowledge distillation" to extract and structure procedural knowledge from LLMs.	Showed that LLMs can be used to generate coherent, step-by-step plans, a process directly analogous to generating recipe instructions.	The work is a general framework for language planning. This project conceptualizes its direct application to the culinary domain for simplifying and personalizing recipe instructions for different skill levels.
Maftei (2025)	Explainability in sustainable food recommender systems.	Proposes a framework for integrating explainable AI (XAI) techniques into	Highlights the importance of transparency and trust in recommender systems, a crucial	The work is a conceptual framework. This project's multi-modal approach, particularly the emotion feature,

		food recommenders.	aspect of user acceptance.	could be a candidate for explainable outputs (e.g., "Recommended because users feel 'love' for this recipe").
Dey & Arefin (2024)	Recommending a household budget.	Rule-based system development.	Addresses the critical real-world constraint of budget, which is often overlooked in academic recommender system research.	The rule-based approach is less flexible than ML models. This project's title acknowledges the importance of budget, proposing it as a key filtering criterion for an advanced ML system.

3 Research Methodology

This research follows a quantitative experimental methodology in designing, implementation and evaluation of the AI meal recommender system proposed. The method is stated in different phases, whereby each phase maps directly to an integral component of the system. The phases in structured approach will guarantee that each part of the system is built and validated solidly before integration into a more comprehensive framework. The methodology focuses on reproducible experiments along with objective, statistical metrics for measuring performance.

3.1 Phase 1: Development of Emotion Classification Model

This introductory phase is all about developing the first model for classification of sentiments. Its main goal is to develop a classifier that would predict the emotion expressed in a specific text.

- Data Gathering: The main corpus collected for this phase consists of the publicly available corpus named the "Emotion Sentiment Dataset," which are text samples and their labels with emotions such as "love", "sadness", "anger", and "neutral".
- Data Preprocessing and Preparation: The textual data is cleared and prepared for modeling through a variety of preprocessing steps: lowercase, removal of special characters, URLs, and common English stopwords elimination. The most significant is balancing the data, which the dataset does show significant imbalance between classes

with a much greater number of samples being classified as neutral than the other major classes, requiring undersampling of the neutral class to almost equal size for modeling purposes to avoid bias towards the larger majority class.

- Model Selection and Training: This task is approached by using a Bidirectional Long ShortTerm Memory (Bi-LSTM) network. Bi-LSTMs are a kind of Recurrent Neural Network (RNN) suited tp sequence modeling tasks as text classification since they can retain and learn contextual information from both preceding and succeeding words in a sentence. Tokenization takes place and padded into a uniform length. This data is fed into the model trained using sparse categorical cross-entropy loss function with the Adam optimizer. An early stop strategy is intended as regularization to avoid overfitting.
- Evaluation: The evaluation of the model can be carried out against a held-out test set that will then be assessed using standard classification metrics such as accuracy, precision, recall, and the F1-score. This would provide a more comprehensive picture of the model's actual ability to correctly identify each class of emotion.

3.2 Phase 2: Model Development for Recipe Preference Score Prediction

This phase employs the trained emotion model from Phase 1 to build her core recommender engine. The prediction needs to meet the scoring quality of recipes based on a multi-modal feature set.

- Data Acquisition and Integration: In terms of datasets, there are two; RAW_recipes.csv is a dataset with complete recipe data (name, ingredients, steps, nutrition), while RAW_interactions.csv contains user ratings and reviews. These datasets are merged using the recipe ID as the main key.
- Data Preprocessing and Sampling: The merged dataset is prepared by cleaning it up by deleting recipes with missing or invalid data such as having a zero cooking time or having no steps. This dataset is a small representation of the whole training dataset based on the continuous user ratings that can then be binned, and thus an equal number of samples per bin can be drawn for the model to have been trained in this very wide spectrum of recipe quality.
- Feature Engineering: This step involves creation of the emotional and nutritional attributes that are needed by the two stages of recommendation pipeline. The Phase 1 emotion classifier is used to process textual input by the user in order to generate an emotion score which is the perceived affective state. This score is an environmental aspect of the following prediction model. Simultaneously, the nutritional characteristics of the recipes such as calories are normalised using Min-Max scaling to make the datasets have identical ranges. These processed features constitute the full input of the preference prediction model, which is trained to approximate the correspondence between each recipe and the emotional state of the user and the calorie limitation requirements indicated by the user.
- Model Selection and Training: An advanced deep learning architecture is designed for this regression task. The attention-based Bi-LSTM network is the model for this study. Well-considered is this architecture since it manages sequence data (the vectorized text)

as well as weighs the importance of its different parts of the input sequence when making a prediction. The model hence works with a joint input combination of the normalized calorie value and continuous score of emotion produced in Phase 1. It is a predictor that is trained to predict a recipe preference score in [01] range with a Mean Absolute Error (MAE) as the optimisation goal.

- Evaluation: The Recipe Preference Prediction model will be assessed using regression metrics on the test set with Mean Absolute Error (MAE) as the primary metric for evaluation, being the most interpretable measure of the average prediction error. Complementary performance picture comes from calculating Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

3.3 Phase 3: SBERT-Based Ingredient Substitution (Conceptual Methodology)

The methodology for this component is clearly laid out, although no implemented code is provided as evidence to examine.

- Model Selection: A pre-trained Sentence-BERT (SBERT) would be chosen. SBERT is a modification of the BERT network where a siamese network structure is used to derive semantically meaningful sentence embeddings compared using cosine similarity.
- Embedding Generation: A comprehensive list of unique ingredients from the recipe dataset would be compiled. Each ingredient name would be fed into the SBERT model to generate a high-dimensional vector embedding. These embeddings would be stored in a searchable index (for example, a vector database, or a simple key-value store).
- Similarity Calculation: When a user requests an ingredient's substitute, its pre-computed embedding would be retrieved. Then cosine similarity would be calculated between this vector and all other ingredient vectors in the index. The ingredients with the highest cosine similarity scores (above a certain threshold) would be considered substitutes and presented.

3.4 Phase 4: Instruction Generation (Conceptual Methodology)

The methodology for the simplification of recipe instructions is likewise described.

- Model Selection: A pre-trained generative language model would be chosen, either T5 (Text-To-Text Transfer Transformer) or a smaller variant of the GPT family. Such models work well in text transformation and generation tasks.
- Task Formulation: The task would be stated as a text-to-text problem. The original recipe instruction step would be the basis for input with a prompt indicating the intended transformation appended (e.g., "Simplify for a beginner: ").
- Inference: The model would process this input and produce a new simplified version of the instruction. This would consist of breaking down complex sentences, defining culinary jargon, or making explicit quantities and times previously conveyed implicitly. This way, the model would exploit the knowledge acquired beforehand and would require little fine-tuning.

4 Design Specification

This section provides architecture system analysis design and special configurations for machine learning models that constitute the main part of the system design. Architectural design modularizes the entire structure, thus allowing component development and test integration.

4.1 System Architecture

The entire system designs a modular architecture pipeline which takes user requests and recipe data and serves personalized recommendations. The architecture consists of some key modules as explained earlier, and as depicted in the Figure 1.

- Data Collection & Storage: The system uses two primary data sources: the emotion sentiment dataset and recipe/interaction datasets, both of which are preprocessed and stored structurally (e.g., with the use of Pandas DataFrames) for easy access during model training and inference.
- Emotion Classification Module: This is an independent module for text-to-emotion prediction. It defines a function which takes an input string of text and returns a predicted emotion label. This module harbors the trained Bi-LSTM model, the tokenizer, and the label encoder.
- Feature Engineering Pipeline: It prepares data for the main recommender model. It processes raw recipe data through a sequence of transformations:
    - Calls the Emotion Classification Module for emotion labels from reviews.
    - Converts the inferred emotion score into a numerical input feature.
    - Normalizes the calorie feature as part of the preprocessing pipeline.
    - Combines the emotion score and calorie value into a unified feature vector.
    - Merges the processed emotion and calorie features into a single model input tensor.
- Recipe Preference Score Prediction Module: This is the trained attention-based Bi-LSTM model and it primarily accepts a processed feature tensor for a recipe, giving a predicted score quality (continuous value between 0 and 1).
- Recommendation Logic Module: Features the user-facing aspect that orchestrates the recommendation process. Receiving user preferences (like some desired emotional quality, nutritional constraints), filtering the recipe database accordingly, sending the filtered candidates through Feature Engineering Pipeline and Recipe Preference Score Prediction Module, and finally targeting the results based on scores to present from them in the top recommendations to the user. Conceptual Modules:
    - Ingredient replacement service. It's going to behave like an API service. It takes in the name of an ingredient and returns a list of the most suitable substitutes guided by the SBERT similarity logic.
    - Instruction Generation Service: Service takes the list of recipe steps and user skill level and uses T5/GPT model to return a modified, simplified

4.2 Data Models

The system is constructed around two major data schemes:

Data on Emotion Sentiment: Simple tabular structure being used. This is summarized in the following Table 3.

Table 3: Schema for the Emotion Sentiment Dataset

Column Name	Data Type	Description
Text	String	Input from the text sample (sentence/tweet).
Emotion	String	Categorical label of the emotion expressed in the text.

Table 4: Processed Recipe Dataset Schema

Column Name	Data Type	Description
recipe_id	Integer	Unique identifier for a recipe.
name	String	Recipe title.
minutes	Integer	Total cooking time (estimated).
ingredients	String	List or string of ingredients in recipes.
steps	String	List or string of cooking instructions.
nutrition	List/Float	Nutritional values (e.g., calories, fat), normalized.
mean_rating	Float	Average user rating for this recipe.
review	String	Example user review for this recipe.
emotion	String	AI-based emotion predicted from review text.

4.3 Algorithm Design

Emotion Classification Model (Bi-LSTM)

The type of this architecture is that of a sequential deep neural network especially for text data.

- Input Layer: It is a sequence of integers of fixed length(maxlen=39), where each integer is representation of a tokenized word from the input text.
- Embedding Layer: An Embedding layer maps these integers into dense vectors of fixed size (100 dimensions). It learns a meaningful vector representation for every word in the vocabulary (input_size=13,530).
- Bidirectional LSTM Layers: Two stacked Bidirectional(LSTM) layers make the core of the model.
- The first Bi-LSTM layer has 128 units and returns the full sequence of outputs. This allows the next layer to receive rich contextual information.
- The second Bi-LSTM layer has 64 units and returns only the final output of the sequence.
- Regularization Layers: In these layers, BatchNormalization and Dropout (with a rate of 0.5) prevent overfitting and enhance generalization. Batch normalization standardizes the inputs to the next layer, while dropout randomly sets a fraction of input units to 0 during training.
- Dense Layers: Two Dense (fully connected) layers follow. The first consists of 64 units with a ReLu activation function and is succeeded by another Dropout layer.
- Output Layer: The last Dense layer has a number of units equals to the number of emotion classes (9) and uses a softmax activation function. Hence, a distribution is produced over the classes, indicating confidence of the model in each possible emotion.

Recipe Preference Score Prediction Model (Attention-based Bi-LSTM)

This model was designed to work with multiple modalities as well as output continuous values.

- Input Layer: The model is built to take in a 3D tensor of shape (batch_size, 100, 319). The dimensions are 100 timesteps (which correspond to the padded sequence length) and 319 features (300 of which come from TF-IDF, 7 are from the nutrition, and 12 from one-hot encoded emotions).
- Normalization and Feature Extraction: LayerNormalization was applied first for stabilization during the learning process. Then, a 1D Conv layer consists of 256 filters and kernel size is 1 then it serves as a very smart dense transformation across the feature vector at each time step so that the model can learn complex interactions among text, nutrition, and emotion features.
- Bidirectional LSTM Layers: The two stacked Bidirectional (LSTM) share units: 64 and 32, respectively, in processing a sequence of feature vectors. Both layers are set to return sequences for a temporal structure preserved for the attention layer downstream.
- Attention Layer: The output of the LSTMs feeds a custom Attention layer that calculates attention weights for each time step in the sequence, thus defining the model's focus on the most relevant parts of the recipe's text and features for prediction. The results of applying it yield a single context vector - a weighted sum of the LSTM outputs.
- Dense Layers for Regression: Context vector goes through these two Dense layers (128 and 64 units) with ReLU activations and Dropout (0.3) as regularization.
- Output Layer: The last Dense layer is a single neuron with a sigmoid activation function. Its application squashes the output to a value between 0 and 1, suitable for normalizing the recipe rating prediction.

5 Implementation

The proposed system is created and implemented through a cloud platform, that is Google Colab, utilizing the programming language Python and its ecosystem of data science and machine learning libraries in Python. Libraries, among others, are provided by TensorFlow (with its Keras API), with Pandas, which is used to manipulate data, Scikit-learn, which handles data preprocessing and evaluation metrics, and NLTK for natural language processing tasks.

5.1 Part 1: Emotion Classifier Model Implementation

The very first step that the emotion classifier was built and trained.

- In this phase, data loading and preliminary cleaning were done: The csv emotion_sentimen_dataset was loaded into a pandas data frame. Already missing emotion labels rows were dropped as the first step to attaining a quality dataset. A simple random sample of 80,000 rows was taken from the dataset to create a smaller yet relevantly large subset for training.
- Data Balancing and Refinement: Class distribution analysis indicated the very severe imbalance of the data, with a dominant 'neutral' class. Consequently, all rare classes (those with very few samples) such as 'surprise', 'empty', 'worry', and 'boredom' were excluded from the complete dataset. The remaining 'neutral' class was then downsized from that originally huge number to a maximum of 3,500 instances. This led to

obtaining a more balanced dataset, which had nine emotion classes - which are very important when training the model for an unbiased classifier.

- Text Preprocessing: A complete text preprocessing pipeline was put in place. A custom function first had to convert the usual internet chat slang (for example, "LOL" to "Laughing Out Loud") to its full form before going for the elimination of any non-alphabetic characters, digits, and excessive whitespaces using regex. This was then followed by converting the whole text to lowercase. Finally, common English stopwords were removed using the NLTK library to reduce noise and refine focus to more meaningful words in the model.
- Data Preparation for Keras: The texts pre-processed from the previous steps (X) and emotional labels (y) were separated from each other. The categorical emotional labels were transformed into integer format using Scikit-learn's LabelEncoder. The dataset was split into training (70%), validation (15%), and test (15%) sets using the stratify option so that the same class distribution was maintained across all sets.
- Tokenization and Padding: A Keras Tokenizer was instantiated and fitted on the training data only, which prevents any leakage of data from the validation or test sets. All three datasets were converted to sequences of integers. The sequences were then padded so they would all fit the longest sequence found in the training data (maxlen = 39), using post-padding to keep uniform input size for the model.
- Model and Training: The Bi-LSTM model, as stated in the design section, was built from the Keras Sequential API. The 'adam' optimizer and 'sparse_categorical_crossentropy' were the loss function with which the model was compiled. An EarlyStopping callback configuration was set up to check on the validation loss and stop training when there are no improvements for three consecutive epochs, reverting weights to the best performing epoch. Then it was trained at a maximum of 30 epochs with a batch size of 64.
- After completing the training of the model, the trained model gets saved into a file called (emotion_bilstm_model.h5). The tokenizer and labelencoder objects were also saved using pickle library in order that the exact same preprocessing can be done during inference.

5.2 Part 2: Recipe Preference Score Prediction Model Implementation

As the second phase of this project, the main recommender engine was constructed.

- Data Loading and Merging: The two datasets, RAW_recipes.csv and RAW_interactions.csv, were loaded. The recipes data underwent cleaning by evaluating the 'nutrition' string into a list and then by expanding this list into separate columns. The datasets were merged on 'recipe_id', and the resultant data were grouped by recipe_id to obtain each recipe's mean rating and one review as a representative review.
- Balanced Sampling: A final working dataset made of 4,000 recipes was created using a balanced sampling approach from the rating bins, ensuring that this model trains over a diverse set of recipes ranging from poorly rated to highly rated, which would avoid any biased tendencies toward a certain tier of quality.

- Feature Engineering with Emotion Model: Saved in Part 1 were loaded, the emotion model, and its relevant preprocessors as well. A method was defined to take a recipe review and clean it through the same preprocessing steps and predict emotion. This method was applied to the 'review' column of the 4,000 sampled recipes to form a new feature column labeled 'emotion'. This column was then label-encoded and one-hot encoded to be used as numerical input for the model.
- Preparing Multi-modal Inputs: The last feature set was prepared.
    - Text Input: The provided text is tokenized with the help of a Keras Tokenizer and transformed into padded integer sequences that can be processed with Bi-LSTM emotion-classification model. The result of this model is a continuous emotion score that is utilized in the next stage- preference-prediction.
    - Numerical Input: The seven nutrition columns were scaled to a range of [0,1] using the MinMaxScaler.
    - Categorical Input: The single numeric score emotion vectors were directly used.
    - In the end, these three feature sets were concatenated along the feature axis to yield a final combined input tensor X_combined. The target variable y was the mean_rating of the recipes, which was also scaled between 0 and 1. The target was then binned into custom buckets to frame the problem like a classification-like regression.
- Model Building and Training: The dataset was split into training and test sets (80/20 split). The attention-based Bi-LSTM model was built as described in the design section. It was compiled with the 'adam' optimizer and an 'mse' loss function. The model was trained for 50 epochs using the combined feature set.
- Recommendation Function: A last function, recommend_recipes, was implemented to encapsulate the entire pipeline of prediction. The function takes a desired emotion and optional nutritional filters as inputs. The master dataset is filtered, and the necessary input tensors for the filtered recipes are prepared. The trained Recipe Preference Score Prediction model is then applied to generate scores, and the top N highest-scoring recipes are returned.

6 Evaluation

This section discusses the very detailed analysis centered on the performances of the two prime machine-learning models developed in this project. Evaluation parameters used on held-out test data provided an unbiased score for each model in terms of effectiveness and generalization capabilities.

6.1 Experiment 1: Emotion Classification Model Performance

Bi-LSTM is emotion classification tandem that acts as the basis for feature engineering pipeline. It is extremely important for the success of the next model which is used in the Recipe Preference Score Prediction. It was evaluated against a test of 2,637 text samples that had neither undergone training nor validation.

The training was monitored to avoid overfitting, as shown in Figure 2. The figures show the accuracy and loss of the model on training and validation datasets in epochs. The validation

accuracy closely tracks the training accuracy before leveling off, and the validation loss reaches a minimum before starting to increase slightly. The early stopping mechanism terminated training at epoch 6, restoring the weights from epoch 3, which had the best validation loss, thus preventing overfitting on training data.

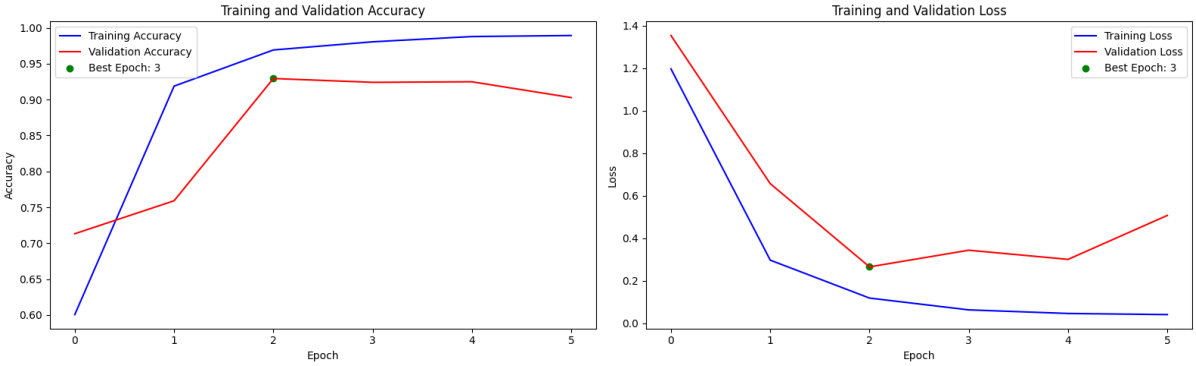

Figure 2: Training and Validation Accuracy and Loss for the Emotion Classification Model

Final performance on the unseen test data yielded 92.87%. It was also recorded that the test loss at this level was approximately 0.263. This further confirmed that the model is very effective in identifying the valid emotion associated with any given text sample.

Table 5: Classification Report for the Emotion Classification Model

Emotion	Precision	Recall	F1-Score	Support
anger	0.99	0.82	0.90	176
enthusiasm	0.80	0.97	0.87	133
fun	0.70	0.95	0.80	143
happiness	0.94	0.96	0.95	382
hate	1.00	0.87	0.93	215
love	0.97	0.98	0.97	575
neutral	0.93	0.88	0.90	525
relief	0.95	0.95	0.95	238
sadness	0.99	0.94	0.97	250
Accuracy	—	—	**0.93**	**2637**
Macro Avg	0.92	0.92	0.92	2637
Weighted Avg	0.94	0.93	0.93	2637

A more granular analysis is forwarded by the classification report in Table 5. This report spells out maximum precision, recall, and F1score for every one of the nine emotion classes.

As seen from these results, generally, performance was very good across the various classes, especially for love, sadness, and relief, with an F1 score for all three equal to or greater than 0.95. Its precision in predicting hate, on the other hand, is a perfect 1.00: when it says hate, it never makes a mistake; however, its recall is a bit weaker at 0.87. The least performance from the model is seen against the metric of precision which is rated at 0.70 and that of F1 which is 0.80 in the context of the fun class, possibly because of some linguistic overlap between the class fun and positive emotions such as happiness and enthusiasm. Nevertheless, the model has shown a fairly high recall value (0.95) for the class of fun, indicating that it has captured most representations of this class correctly. The weighted average F1 score of 0.93 corroborates the overall robustness of the model.

6.2 Experiment 2: Recipe Preference Score Prediction Model Performance

This second experiment discusses the attention-based Bi-LSTM model designed for predicting the recipe quality score. The performance of this model needs to be closely related to the quality of the subsequent recommendations provided in the end. The model was evaluated through a test consisting of 800 recipes out of a 4000-sample dataset (20% testing).

The training of the model continued for 50 epochs. An analysis of the training history indicated that validation loss would initially decrease, plateau, and then slightly increase, which indicated the beginning of overfitting. The performance of the model on this test set must be the one just before major overfitting occurred.

Main evaluation metrics for this regression task are presented in Table 6. Predictions of the model (y_pred_raw) are now continuous values between 0 and 1. In order to make a more reliable evaluation, those raw predictions will be then bucketized to the nearest valid target value (0.1, 0.2, 0.4, 0.6, 0.7) prior to calculating metrics such as MSE and MAPE. Direct Mean absolute error (MAE) can be calculated on these raw predictions.

Table 6: Evaluation Measures for Prediction Model of Recipe Ratings

Metric	Value	Description
Mean Squared Error (MSE)	0.0592	Average of the squares of prediction error.
Root Mean Squared Error (RMSE)	0.2433	Root of the mean square error computed from prediction.
Mean Absolute Error (MAE)	0.1890	Average difference between observed and forecast values; common for single-observer measures.
Mean Absolute Percentage Error (MAPE)	0.7047	Mean of absolute percentage errors.
Median Absolute Error (MedAE)	0.2000	Median of all absolute differences; robust against outliers.

Thus, the most interpretable metric in the equation is the Mean Absolute Error (MAE), which stands at 0.1890. Given that the ratings were transformed into a range of [0, 1] (essentially a 1 to 5-star scale), this MAE typifies an average prediction error of roughly 0.189 * 4 + 1, which is below one star on the original 5-star scale. In fact, approximately about 0.945 stars. Although that may appear somewhat high, it should be pointed out that this task is complex in nature. Predicting an inherently subjective measure like recipe rating from text and numbers is difficult. The Median Absolute Error of 0.2 is somewhat similar; under this, half the estimates were less than this value in deviation. The relatively high MAPE figure is skewed toward predictions for lesser-rated recipes, where a little amount of absolute error can produce a comparatively larger percentage error.

6.3 Discussion

The evaluative results of both experiments are very encouraging and also substantiate the basic assumptions made in this study.

The finding with respect to the success of the emotion classification model (Experiment 1)-the accuracy was 92.87%-was indicative that a properly tuned Bi-LSTM was already more than

capable of capturing the emotional undertones of texts written by users. The high level of performance was critical because the output of this model became one of the two most important input features feeding into the Recipe Preference Score Recommender. Any substantial noise in error coming from an inaccurate emotion model would cause serious degradation in the performance of the downstream task.

The evidence for the validity of the multi-modal approach proposed is very strong; really, the second experiment's performance put the recommender of recipe in a very impressive light. Then, the model's achievement of an MAE of 0.189 demonstrated that recipes would be predicted value by combining recipe textual information, nutrient data, and most importantly, the AI-derived emotional sentiment. The attention mechanism was probably instrumental in allowing the model to concentrate on the most salient features within the long ingredient and instruction text.

The single review would be randomly chosen for each recipe, which is a major weakness in the evaluation data, and those would be the emotion-the feature measure comes from that single review. It would be better to combine several reviews' emotions for a more stable emotional profile. Also, the subjective nature and noise in user ratings place a ceiling on what could be done by any prediction model. For example, ratings given by end users may be caused by biases outside what is in the data, such as by individual differences in cooking mistakes or differences in ingredient quality.

Even though such limitations are obvious, this evaluation shows that the two-stage pipeline is an apt and effective methodology to develop more complicated meal recommenders. Accordingly, this evaluation narrative will justify the design choices for future work put down in the next section clearly.

7 Conclusion and Future Work

This research project aims to build and develop an advanced AI-based meal recommender system intended to mitigate some of the chief shortcomings of existing platforms. The purpose was mainly to analyze the practicability of a multi-stage machine learning pipeline for rating prediction, together with paving the way for intelligent ingredient substitution and a generative instruction scheme.

These thorough investigations more than suffice to address the first research question. A rigorous evaluation of the two-stage pipeline was developed and conducted. Stage one designed and implemented an accurate Bi-LSTM model for emotion classification from user reviews, which achieved a stellar 92.87% on unseen data. Stage two coupled the emotion classifier to act as a feature engineering variable to a new attention-enhanced Bi-LSTM rating prediction model. By integrating textual, nutritional, and emotional variables, this model showed fantastic validation results of 0.1890 MAE on a scaled rating system. Accordingly, these insights validate that sentiment discerned via AI would be an excellent feature to predict recipe quality, one that offers richer signal value than explicit star ratings alone.

In addition, the project therein gives a very clear description of how to go about addressing the remaining second and third research questions covering SBERT-based ingredient substitution and generative instruction simplification. While those components were not implemented, their proposed designs are based on state-of-the-art techniques and seem to be a logical and viable extension to the core system.

This research's primary implication is that it charts a pathway to smarter, contextually cognizant, and more invaluable culinary aides. Instead of being based on trivial content-based

or collaborative filtering it recommends based on a nuanced understanding of a recipe's attributes including its possibly emotionally eliciting dimensions. A huge drawback that hampers our present work concerns the non-implementation of the budget, substitution, and instruction generation modules. On top of that, the performance of the system is pretty much contingent on the quality and availability of user reviews for its feature generation.

7.1 Future Developments

Implementation of the core recommendation engine has embarked on very promising upshots in further investigations and developments.

- Full System Implementation: Among the very next steps to be undertaken are those of installing the modules conceived. These include the construction of the SBERT-based service for ingredient substitution and the generative model for instruction simplification. Integration of a price estimation model for ingredients would satisfy the "Budget-Based" aspect of the project's title and would render the system complete.
- More Personalization: The current system recommends recipes mostly used within a given emotion. An important step would be to incorporate user profile specific attributes such as diet preferences, allergies, previous recipes selected, and explicit feedback, thus rendering truly personalized recommendations.
- Thus, Recipes are noted to be multi-modal. The further versions of the model could utilize images as another input. CNN could be leveraged to extract features from the recipe images, which could thus provide strong signals for the dish's attractiveness and complexity.
- Dynamic and Interactive Instruction Generation: Evolve the instruction-generating module from a static simplifier to an interactive conversational agent. So, could the user ask clarifying questions throughout cooking: "How do I know when the onions are nicely caramelized?" or "What can I use instead of a whisk?" And the system would answer in real-time.
- Commercialization Potential: The developed system has solid commercial potential. It could be implemented either as a standalone mobile application, as a web service, or an API to be licensed to grocery delivery services, manufacturers of smart kitchen appliances, or digital content creators in the food industry.

In summary, this project has successfully shown the advantage of integrating multiple inter-advanced machine learning models to materialize a rather sophisticated and competent meal recommendation engine. The validation of the two-step pipeline stands as an undoubted strong contributor to the field and presents a great and sturdy springboard for the next generation of intelligent culinary tools.

References

Alfifi, M., Dong, X., Feldman, T., Lin, A., Madanagopal, K., Pethe, A., Teleki, M., Wang, Z., Zhu, Z. and Caverlee, J., Howdy Y'all: An Alexa TaskBot.

Boukhatem, M., 2024. *Natural Language Processing Approaches for Closed-Domain Technical Dialogues* (Doctoral dissertation, Institut Polytechnique de Paris).

Chen, F., Bedmutha, M.S., Chung, R.Y., Sabin, J., Pratt, W., Wood, B.R., Weibel, N., Hartzler, A.L. and Cohen, T., 2025, May. Toward Automated Detection of Biased Social Signals from the Content of Clinical Conversations. In *AMIA Annual Symposium Proceedings* (Vol. 2024, p. 252).

Demir, S., Fuegener, A., Gupta, A. and Weinmann, M., 2024. The effect of AI support on Torrance's creativity dimensions: Evidence from an online experiment.

Dey, S. and Arefin, M.S., 2024. Developing a Rule-Based System to Recommend Household Budget. *The Lancet*, *403*(10423), pp.10-1016.
Fabris, F., 2024. Reimaging profiling: LLMs for next era of hyper-personalization.
Jouravel, A., 2025, January. Retrieval of Parallelizable Texts Across Church Slavic Variants. In Proceedings of the 12th Workshop on NLP for Similar Languages, Varieties and Dialects.
Kamra, V.V.F.X.N., Matias, P., Carlberg, K. and Ridgeway, J.T.K., 2023. Benchmarking Egocentric Multimodal Goal Inference for Assistive Wearable Agents. Science, 9, p.00.
Kim, H., Venkataramanan, R. and Sheth, A., 2024. A Survey on Food Ingredient Substitutions. arXiv preprint arXiv:2501.01958.
Kopitar, L., Bedrač, L., Strath, L.J., Bian, J. and Stiglic, G., 2025. Improving Personalized Meal Planning with Large Language Models: Identifying and Decomposing Compound Ingredients. Nutrients, 17(9), p.1492.
Maftei, C., 2025. Developing a Framework for Integrating Explainability for Sustainable Food Recommender Systems (Bachelor's thesis, University of Twente).
Morales-Garzón, A., Gómez-Romero, J. and Martín-Bautista, M.J., 2022, July. Contextual sentence embeddings for obtaining food recipe versions. In International conference on information processing and management of uncertainty in knowledge-based systems (pp. 306-316). Cham: Springer International Publishing.
Mereddy, D. and Beedareddy, J.S.R., 2024, December. Enabling Next-Generation Smart Homes Through Bert Personalized Food Recommendations-RecipeBERT. In 2024 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT) (pp. 796-803). IEEE.
Senath, T., Athukorala, K., Costa, R., Ranathunga, S. and Kaur, R., 2025. Large language models for ingredient substitution in food recipes using supervised fine-tuning and direct preference optimization. Natural Language Processing Journal, p.100177.
Sha, H., Gong, F., Liu, B., Liu, R., Wang, H. and Wu, T., 2025. Leveraging retrieval-augmented large language models for dietary recommendations with traditional Chinese Medicine's medicine food homology: algorithm development and validation. JMIR Medical Informatics, 13(1), p.e75279.
Shirai, S.S., Seneviratne, O., Gordon, M.E., Chen, C.H. and McGuinness, D.L., 2021. Identifying ingredient substitutions using a knowledge graph of food. Frontiers in Artificial Intelligence, 3, p.621766.
Suresh, S.K., Surana, T., Hao, L.Z. and Chng, E.S., 2025. CS-Sum: A Benchmark for Code-Switching Dialogue Summarization and the Limits of Large Language Models. arXiv preprint arXiv:2505.13559.
Veerabadran, V., Xiao, F., Kamra, N., Matias, P., Chen, J., Drooff, C., Roads, B.D., Williams, R., Henderson, E., Zhao, X. and Carlberg, K., 2025. Benchmarking Egocentric Multimodal Goal Inference for Assistive Wearable Agents. arXiv preprint arXiv:2510.22443.
Wang, J.J., 2025. Building Recommender Systems Using Large Language Models.
Yuan, S., Chen, J., Fu, Z., Ge, X., Shah, S., Jankowski, C., Xiao, Y. and Yang, D., 2023, July. Distilling script knowledge from large language models for constrained language planning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 4303-4325).