SNACK HUNTER

THE ULTIMATE TAIWANESE NIGHT MARKET FOOD FINDER

四234 - GROUP 9 - MIS

112306079 彭宇承 112306092 許新翎 112306093 林秉毅
112306018 陳詠旭 112701030 陳允中

1131 Data Structure – Final Project Proposal

四 234 – Group 9 – MIS

112306079 彭宇承 112306092 許新翎 112306093 林秉毅 112306018 陳詠旭 112701030 陳允中

Final Project – Let's Beat Google!

I. Introduction

A. Topic

Snack Hunter: The Ultimate Taiwanese Night Market Food Finder

B. Motivation

In this rapidly evolving technological era, people are increasingly relying on the convenience brought by advanced technologies. However, effectively leveraging these tools has become a critical challenge. Taiwan's night markets are globally renowned for their diverse and unique culinary delights. Yet, for international visitors, language barriers, religious dietary restrictions, or personal preferences often make it difficult to accurately identify the ingredients and reviews of the food in front of them.

To address this issue, we have developed a unique search technology that integrates language translation with artificial intelligence. Our system intelligently recognizes user input keywords and food descriptions, converting them into optimized keywords suitable for Google searches. By leveraging Google search results, combined with our extensive database of Taiwan night market food-related keywords, the system maps, re-scores, and reorders the results to enhance their accuracy.This technology empowers travelers from around the world to overcome language and cultural barriers, enabling seamless exploration of Taiwan's diverse night market cuisine. Not only does it allow visitors to enjoy local delicacies with greater confidence, but it also significantly enhances Taiwan's cultural visibility and accessibility on the global stage.

C. Why can we beat Google? (Our Advantage)

1. **Multilingual Processing**

   When searching for "Taiwan night market food" using foreign languages, the results are often irrelevant or entirely absent. To address this, we propose a "query preprocessing" framework. By leveraging Large Language Model (LLM) technology, our system maps user queries to our curated keyword set, identifying all core concepts the user intends to express. These are then translated into Chinese for the search process, dramatically improving accuracy compared to Google's native search capabilities for non-Chinese languages.

2. **Natural Language to Keywords Conversion**

   Using CKIP Transformer technology, we preprocess user input to convert both natural language queries and keyword-based questions into a structured keyword format. This approach allows for flexibility in input styles, eliminating the need for users to follow Google's rigid keyword-based query requirements.

3. **Relevance Reordering with Localized Weighting**

   Our system crawls and analyzes a vast collection of Taiwan night market-related keywords, applying a weighted scoring mechanism to re-rank Google search results. Unlike Google's generic approach, which lacks customization for niche topics like Taiwan night market cuisine, our localized keyword database enables enhanced relevance. This ensures that search results are significantly more accurate and tailored to the user's intent.

II. Search Tricks

A. Ranked Websites *(Examples provided here; refer to the section below for the "Complete URL Collection")*

- https://www.gvm.com.tw/article/68042
- https://tw.my-best.com/114658
- https://www.travelking.com.tw/tourguide/nightmarket/snacks.asp

- https://home.csulb.edu/~jwinter2/chin410/projects2004/ilee/nightmarket.html
- ... ...

Complete URL Collection：

https://docs.google.com/spreadsheets/d/1NcjCGWcwnUiV9Bc6aELdn4uqNFMNJ6J6oJY−UAdL2Go/edit?usp=sharing

B. Keywords *(This is a partial example; additional keywords will be extensively crawled using the program.)*

Complete Keyword Collection:

https://docs.google.com/spreadsheets/d/1NcjCGWcwnUiV9Bc6aELdn4uqNFMNJ6J6oJY−UAdL2Go/edit?gid=960021094#gid=960021094

County/City：臺北、高雄、臺中、...

Night Market Name：士林夜市、逢甲夜市、饒河夜市、...

Food Type：小吃、燒烤、甜點、飲料、...

Food Name：鹽酥雞、臭豆腐、大腸包小腸、...

Keyword Weighting:

County/City：5.0

Night Market Name：10.0 (Usually the Most Accurate)

Food Type：8.0 (Confirm Food Type)

Food Name：8.0 (Find Specific Food Name)

C. Score Formulation

1. **Keyword Matching Scoring**

   When the user's query keywords match the content on a website, a base score is assigned based on the frequency of the keywords

appearing on the site. This score is adjusted according to the degree of keyword match.

Unlike assigning individual weights to single keywords, our keyword matching scoring system assigns fixed weights based on keyword categories (e.g., county, night market name, etc.).

2. **Weighted Calculation**

Total Score = (Number of County Matches × 5) + (Number of Night Market Name Matches × 10) + (Number of Food Type Matches × 8) + (Number of Food Name Matches × 8)

III. System design

A. System Workflow

1. Question Analysis and Keyword Generation

1.1 Question Content Check

When the user's query contains characters other than "Chinese," "numbers," or "punctuation marks," the system activates the GPT API for analysis:

1.1–1 GPT Content Analysis

The system sends the "user query" and "[pre–selected keywords](#)" to the GPT API to identify potential keywords related to the query.

1.1–2 Keyword Integration

The results are consolidated into a simplified, space–separated input format to improve processing accuracy.

Example:

- Original Query: *What are the popular barbecue options at Taiwan night markets?*
- Converted: *臺灣 夜市 烤肉*

1.2 Keyword Segmentation Processing

If the user's query contains only "Chinese," "numbers," or "punctuation marks," the system utilizes the CKIP Transformers tokenizer for keyword generation:

- The tokenizer converts natural language questions into keyword format, ensuring the query is prepared for the search engine in standard format.
- Example:

  Original Query: *我想尋找臺灣夜市最讚的烤肉店*

  Tokenized: *尋找 臺灣 夜市 烤肉 店*

2. Keyword Search and Ranking

Regardless of whether 1.1 or 1.2 is triggered, the user's query is eventually converted into "keyword format" and enters the search phase:

2.1 Website Weight Calculation

Based on a predefined list of websites (refer to [II. Search trick – A. Ranked Websites](#)), the system calculates the relevance score for each website using the weighted scoring method defined in HW6.

2.2 Sorting and Output

The results are ranked based on the calculated scores and the recommended websites are displayed on the front–end interface (using HW7's ranking methodology).

3. Google API Deep Search

When the user clicks the **Search** button on the front–end, the system activates the Google API for further exploration:

3.1 Extract Google Search Results

The system retrieves the top 20 web pages from Google search results and applies the same weighted scoring method as in 2.1 to generate a ranked output.
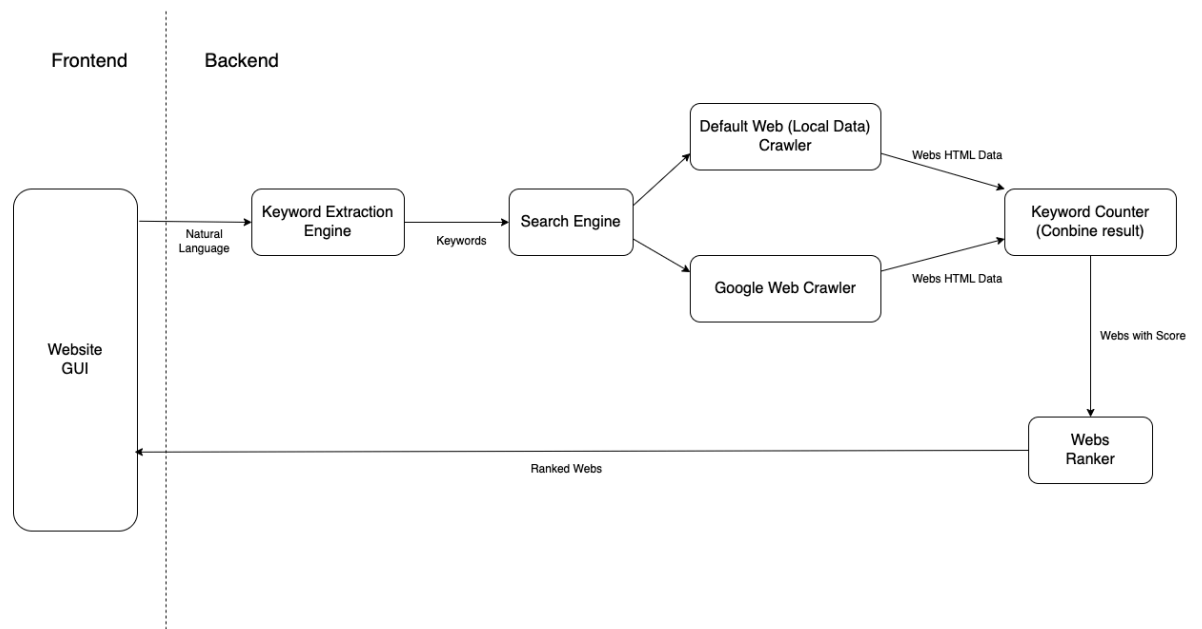
3.2 Merged Ranking Results

Google search results are merged with the rankings from **2.** for

local websites. If duplicate web pages exist, only one is retained (with the same score). The final result is output to the front-end interface.
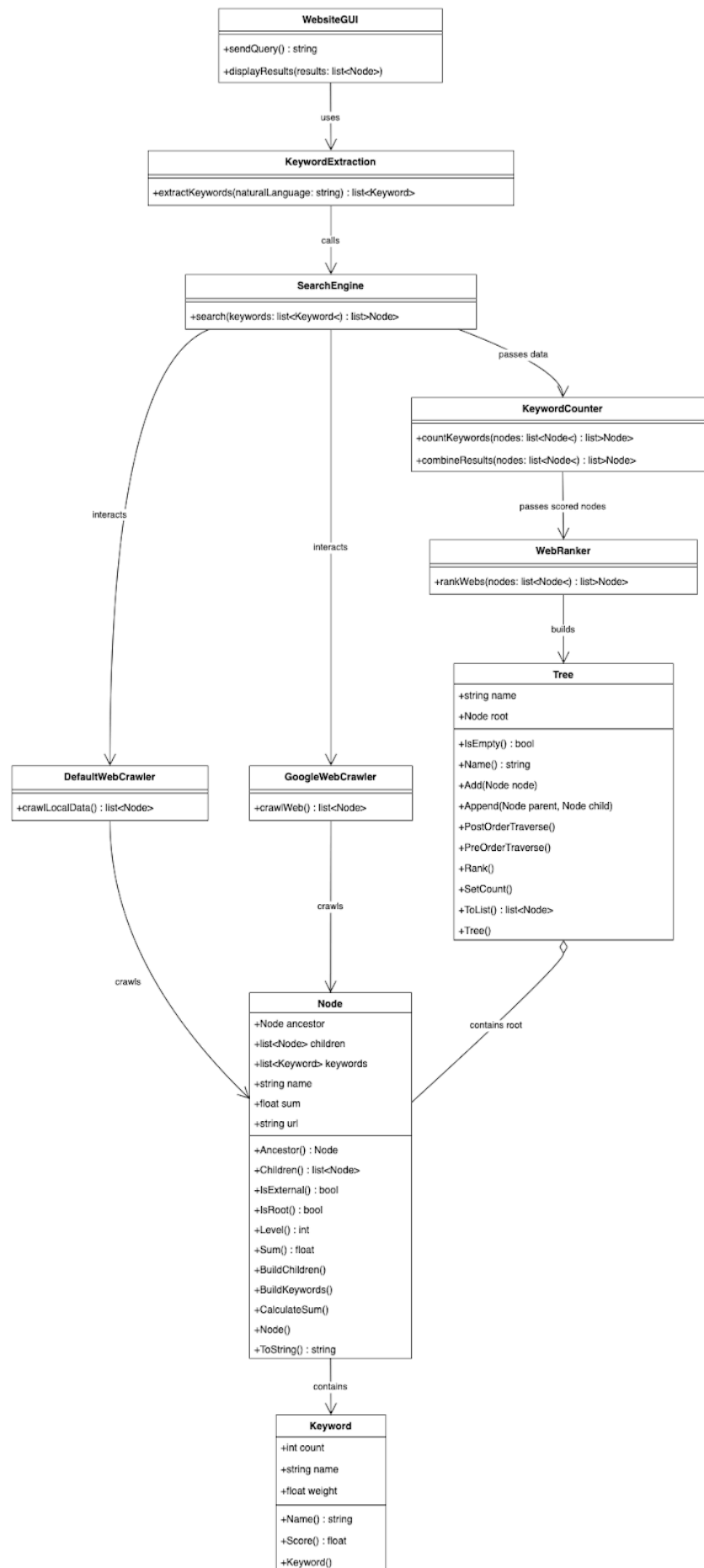
B. Data flow diagram
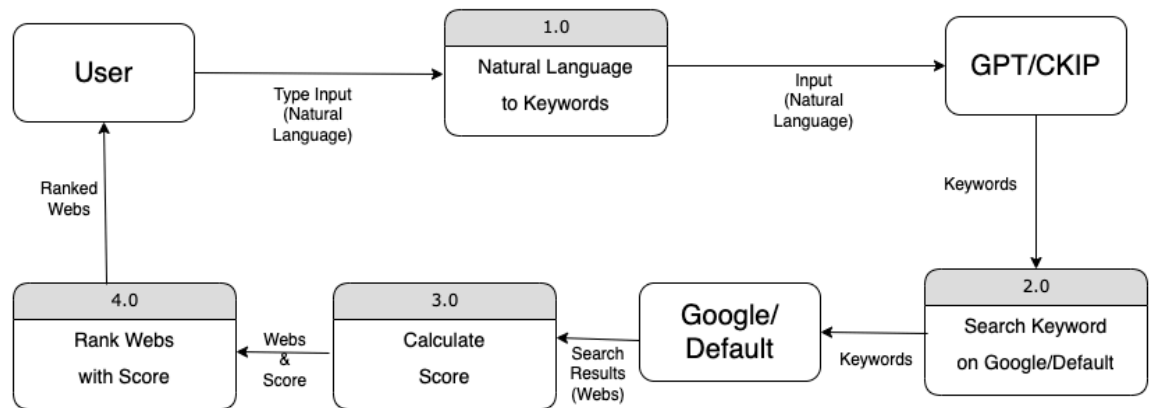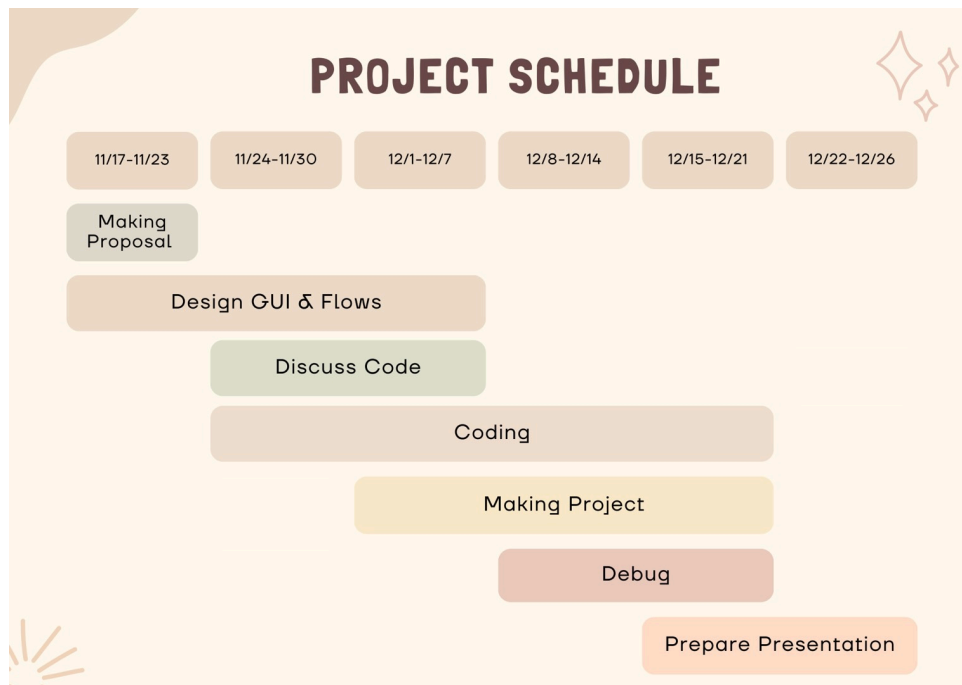

C. System structure diagram

D. Class diagram (Initial planning; subject to iterative adjustments based on actual development.)

WebsiteGUI

+sendQuery() : string

+displayResults(results: list<Node>)

uses

KeywordExtraction

+extractKeywords(naturalLanguage: string) : list<Keyword>

calls

SearchEngine

+search(keywords: list<Keyword<) : list>Node>

passes data

KeywordCounter

+countKeywords(nodes: list<Node<) : list>Node>

+combineResults(nodes: list<Node<) : list>Node>

passes scored nodes

WebRanker

+rankWebs(nodes: list<Node<) : list>Node>

builds

Tree

+string name

+Node root

+IsEmpty() : bool

+Name() : string

+Add(Node node)

+Append(Node parent, Node child)

+PostOrderTraverse()

+PreOrderTraverse()

+Rank()

+SetCount()

+ToList() : list<Node>

+Tree()

interacts

interacts

DefaultWebCrawler

+crawlLocalData() : list<Node>

GoogleWebCrawler

+crawlWeb() : list<Node>

crawls

crawls

contains root

Node

+Node ancestor

+list<Node> children

+list<Keyword> keywords

+string name

+float sum

+string url

+Ancestor() : Node

+Children() : list<Node>

+IsExternal() : bool

+IsRoot() : bool

+Level() : int

+Sum() : float

+BuildChildren()

+BuildKeywords()

+CalculateSum()

+Node()

+ToString() : string

contains

Keyword

+int count

+string name

+float weight

+Name() : string

+Score() : float

+Keyword()

IV. Schedule

V. Challenges

A. Translation Errors

Due to cultural differences, a single word may refer to different foods in different regions. For instance, "chips" in British English means "fries," while in American English, it refers to "potato chips." If a user's search query contains such terms, it may result in incorrect outputs.

B. Precision of Word Usage

If users are unsure how to describe the snack they wish to explore and use vague terms (e.g., color, shape, flavor), the search engine may output results that match the characteristics but may not align with the exact snack the user is looking for.

C. Search Feedback Speed

To enhance the search accuracy for this specialized topic (Taiwan night market cuisine), we integrated CKIP Transformers for tokenization and LLM–based multilingual keyword extraction. However, combining these processes with the Google API may result in longer execution times for a single search, potentially affecting the user experience.