

SpotLite 簡介

[Copy](#)

SpotLite: 基於面向分析與個人化排序的旅遊景點周邊餐廳推薦系統 (SpotLite: A System for Personalized, Aspect-Based Restaurant Recommendation Near Tourist Destinations)

1. 動機與問題定義 (Motivation & Problem Statement)

現有的餐廳搜尋平台（如 Google Maps, Yelp）主要依賴 **星等 + 零散關鍵字** 的模式，但這種模式對於有特定需求的旅行者存在三大痛點：

1. **資訊過載 (Information Overload)**：使用者必須逐一點開並閱讀大量評論，才能掌握「環境是否安靜」、「價格是否合理」等特定面向的真實情況，決策效率低下。
2. **關鍵字零散 (Fragmented Keywords)**：語義相似的詞彙（例如：便宜、平價、不貴、CP值高）不會被自動整合，導致使用者搜尋時資訊獲取不完整，難以全面理解。
3. **缺乏深度個人化 (Lack of Personalization)**：現有平台的篩選功能過於基礎，無法支援「距離近、適合約會、但要避開排隊人潮」這類結合了多維度條件的複雜查詢。

我們的解決方案：SpotLite

為了解決以上痛點，我們提出 **SpotLite** 系統。使用者只需輸入一個 **Google Maps** 的景點分享連結，系統即可自動推薦附近餐廳，並輸出一份包含 **多面向分析、相似關鍵字聚類、AI 摘要**，以及**個人化排序** 的深度報告。

功能	Google Maps	SpotLite (本專案)
評分排序	✓	✓
零散關鍵字搜尋	✓	✗ (自動聚類整合)
面向分析	✗	✓
AI 生成摘要	✗	✓
多維度個人化偏好	✗	✓

專案範疇：為確保專案聚焦與模型品質，初期將僅處理過去一年內的英文評論。

2. 相關研究 (Related Work)

我們的系統建立在 foundational NLP 模型與演算法之上，特別是利用以 BERT 為基礎的預訓練 Transformer \cite{devlin2018bert} 來進行深度語言理解。

- **面向情感分析 (ABSA)：**我們將遵循該領域的經典方法 \cite{hu2004mining} 來實現面向抽取與情感分類，這是我們系統分析使用者評論的核心。
- **關鍵字抽取 (Keyword Extraction)：**我們將採用基於圖的 TextRank 演算法 \cite{mihalcea2004textrank} 來識別核心關鍵字，此方法比單純的詞頻模型更為穩健。
- **語義聚類 (Semantic Clustering)：**為了將同義詞分組，我們將使用 Sentence-BERT \cite{reimers2019sentence} 來計算語句嵌入向量，從而實現語義層面的關鍵字整合。
- **摘要生成 (Summarization)：**我們將使用一個經過微調的 T5 (Text-to-Text Transfer Transformer) 模型 \cite{raffel2020exploring} 來執行基於面向與關鍵字的抽象式摘要，以生成流暢自然的總結。

3. 系統設計與架構 (System Design & Architecture)

3.1 系統處理流程 (Pipeline)

Google Maps Link → Geocoding → Nearby Search → Reviews
↓ 清洗 & 去重
↓ 面向分類 (Aspect Detection)
↓ 關鍵字抽取 & 相似詞聚類
↓ 情感分析與量化 (Sentiment Quantification)
↓ AI 摘要 (Generative Summarization)
↓ 個人化篩選與排序 (Preference-aware Ranking)
→ 輸出 JSON / PWA 卡片

3.2 核心技術元件細節

1. 資料前處理：

- **工具：** `langdetect`, SimHash, MiniLM embeddings。
- **功能：** 過濾非英文評論；使用 SimHash 進行句法去重，再用 MiniLM 語義嵌入向量進行語義去重，確保資料品質。

2. 面向分類 (Aspect Detection)：

- **分類體系 (Taxonomy)：** 口味 (Taste), 價格 (Price), 服務 (Service), 環境 (Ambiance), 等候時間 (Wait Time), 交通 (Location), 菜系 (Cuisine), 特殊需求 (Special Needs)。
- **方法：** 採用半監督學習，使用一個種子詞典 (seed dictionary) 進行初步標註，再透過 embedding 相似度擴展詞彙，以捕捉未登錄的同義詞。

3. 情感分析與量化：

- **分析單位：** 在句子層級分析每個面向的情感 (正面/負面)。
- **量化統計：** 匯總統計過去一年的 `positive_mentions` 和 `negative_mentions`。
- **關鍵字歸因：** 將與各情感相關的關鍵字分別收集到 `positive_keywords` 和 `negative_keywords` 列表中。
- **量化分數 (score)：** 為每個面向計算一個標準化的綜合分數，公式可設計為： $Score_{aspect} = \sigma \left(w_1 \cdot \log(N) + w_2 \cdot \overline{|S|} \right)$ 其中 N 為總提及次數， $\overline{|S|}$ 為平均情感強度， w_1, w_2 為權重。

4. AI 摘要與個人化排序：

- **摘要生成**：一個 **t5-small** 模型根據分數最高的幾個面向及其情感關鍵字，生成 2-3 句的摘要。
- **個人化排序**：最終排名由一個加權函數決定，綜合考量餐廳評分、評論數、距離、面向分數和使用者的偏好匹配度。
$$final_score = \alpha \cdot rating + \beta \cdot \log(1 + reviews) - \gamma \cdot distance + \delta \cdot pref_boost$$

3.3 系統 API 輸入/輸出範例

輸入 (Input - JSON)

```
{
  "location_url": "[https://share.google/nWiVMXDYMuUHvmxte]
(https://share.google/nWiVMXDYMuUHvmxte)",
  "search_radius_meters": 800,
  "preferences": {
    "must_have": ["Japanese Cuisine"],
    "nice_to_have": ["quiet", "priced"],
    "exclude_keywords": ["queue", "noisy"]
  }
}
```

輸出 (Output - JSON)

```
[
  {
    "rank": 1,
    "name": "The Tuna Sushi",
    "ai_summary": "A highly-rated spot known for its fresh fish.
The ambiance is consistently described as quiet, making it ideal
for dates. While praised for quality, some note the prices are on
the higher end.",
    "aspect_analysis": [
      {
        "aspect": "Taste",
        "score": 0.94,
        "positive_mentions": 85,
        "negative_mentions": 5,
        "positive_keywords": ["fresh fish", "quality tuna"],
        "negative_keywords": ["rice was mushy"]
      },
      {
        "aspect": "Ambiance",
        "score": 0.87,
        "positive_mentions": 65,
        "negative_mentions": 10,
        "positive_keywords": ["quiet", "intimate"],
        "negative_keywords": ["a bit dark"]
      }
    ],
    "pros": ["High-quality ingredients", "Quiet atmosphere"],
    "cons": ["Higher price point"]
  }
]
```

4. 研究假說與評估方法 (Hypothesis and Evaluation)

研究假說：一個整合了面向特徵抽取和多標準偏好排序的系統，在推薦的「相關性」(以 NDCG 衡量) 和「使用者效率」(以任務完成時間衡量) 上，將比依賴綜合評分的基線系統有顯著的提升。

我們將進行一個雙層的量化評估：

4.1 模組層級驗證

- 面向分類：在一個手動標註的資料集 (N=200) 上，F1-score 需 ≥ 0.75 。
- 關鍵字聚類：與一個標準同義詞集合比對，準確率需 $\geq 80\%$ 。

4.2 端到端系統評估

- 排序相關性：與 Google Maps 的預設排序相比，NDCG@10 指標需提升 $\geq 15\%$ 。
- 使用者效率研究：一個對照實驗，用以衡量任務完成時間 (目標：縮短 $\geq 50\%$) 和回答正確率 (目標：與對照組的差距在 10% 以內)。

5. 開發時程與交付成果 (Timeline & Deliverables)

專案預計在四周內完成，各階段里程碑與人員分工如下：

週次 (Week)	里程碑 (Milestone)	交付成果 (Deliverables)	負責人 (建議)
W1	資料獲取與前處理	能穩定輸出乾淨 Top-20 評論的 JSON	組員 A (API), B (清洗)
W2	核心 NLP 模組 (I)	每則評論完成面向/情感/關鍵字標註	組員 C (分類), D (抽取)
W3	核心 NLP 模組 (II)	輸出包含摘要與個人化排序的完整 JSON	組員 E (聚類/排序), F (摘要)
W4	系統整合與評估	可運行的 PWA Demo 與最終評估報告	全員

最終交付成果

1. **後端 API**：輸入景點連結，輸出包含所有分析的 JSON。
 2. **前端 Demo**：一個手機友善的 Web App (PWA) 用於展示。
 3. **專案報告**：包含所有流程圖、評估結果與案例分析。
 4. **展示影片**：1-2 分鐘的螢幕錄影，展示完整使用流程。
-

6. 結論與未來展望 (Conclusion & Future Work)

本專案結合了 NLP、搜尋體驗優化與量化驗證，旨在打造一個真正能解決旅客痛點的智慧推薦系統。

未來應用：

- 擴展至 **酒店推薦** (例如：交通便利、安靜、早餐好)。
 - 擴展至 **旅遊活動推薦** (例如：適合家庭、體驗獨特、CP值高)。
 - 最終目標是成為一個完整的 **旅遊智慧推薦解決方案**。
-

7. 參考文獻 (References)

- Devlin, J., et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT*.
- Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing Order into Texts. *EMNLP*.
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *EMNLP*.
- Raffel, C., et al. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*.
- Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *KDD*.

Last updated 0 minutes ago