SpotLite: A System for Personalized, Aspect-Based Restaurant Recommendation Near Tourist Destinations

Ting-Long Wei twei0542@usc.edu

Yu-Chen Lu ylu74747@usc.edu

Sheng-Kuo Lin slin7099@usc.edu

Shih-Hui Huang shihhuih@usc.edu

Jingyi Xia xiajingy@usc.edu

Suihan Gao suihanga@usc.edu

University of Southern California, Los Angeles, CA, USA

Abstract

Online platforms like Google Maps contain rich restaurant reviews but provide limited insight at the aspect-level and unclear sentiment, making it difficult for users to identify what truly matters. We present SpotLite, an NLP system that converts unstructured reviews into structured evaluations across Taste, Service, Environment, Waiting Time, and Price. Our pipeline combines MiniLM-based aspect clustering, DistilBERT sentiment analysis, TF-IDF keyword extraction with co-occurrence clustering, and a fine-tuned T5 model for aspect-aware summarization. The LLM-based evaluation shows strong aspect accuracy and keyword quality, with remaining challenges in summary richness. SpotLite offers an interpretable approach to large-scale restaurant understanding.

1 Introduction

Diners rely on platforms such as Google Maps to choose restaurants, yet these platforms present three major challenges: information overload, lack of aspect-level insights, and unclear sentiment. Users must manually scroll through long, unstructured reviews without clear guidance on taste, price, or other factors that matter to their decisions.

To address these limitations, we introduce SpotLite, a system that turns restaurant reviews into structured, aspect-aware insights. It aims to answer three key questions: 1) How can unstructured text be transformed into interpretable aspect categories? 2) How can sentiment be assigned reliably across aspects? 3) How can we generate concise summaries and keywords that reflect customer experiences?

Our pipeline integrates embedding-based aspect clustering, sentiment analysis, keyword extraction, and generative summarization to produce interpretable restaurant profiles.

2 Related Work

2.1 Aspect-based Sentiment Analysis

Carrasco and Dias propose an automated aspect-based sentiment analysis (ABSA) pipeline for restaurant reviews that classifies sentiment by operationally meaningful attributes (e.g., food quality, service, ambiance, pricing, and location) and deploy it in a web application to support monitoring and competitor comparison (Carrasco and Dias, 2024). Similar to their goal of extracting aspect-level insights from large-scale online reviews, our system also structures review text into aspect-specific signals. However, while their study emphasizes supervised attribute extraction and sentiment classification for managerial analytics, our work prioritizes interpretable aspect clustering, keyword extraction, and aspect-aware summarization aimed at user-facing understanding and transparent evidence presentation.

2.2 Review Generation

Maldonado Castillo et al. (Maldonado Castillo et al., 2024) propose a system that aggregates reviews from multiple platforms to generate comprehensive restaurant-level summaries using supervised aspect–opinion–sentiment triplet extraction and LLM-based generation. While both their work and ours aim to reduce information overload by synthesizing large-scale restaurant reviews, their approach focuses on supervised triplet modeling and fully generative reviews. In contrast, our system emphasizes unsupervised aspect clustering, lightweight sentiment analysis, and evidence-grounded summaries to improve interpretability and transparency.

2.3 Emotion Recognition

Liu et al. (Liu et al., 2024) study emotion recognition in restaurant online reviews using a hybrid model that integrates deep learning with a sentiment lexicon, demonstrating strong performance on both valence-based and discrete emotion classification at scale. While their work focuses on fine-grained emotion detection for service quality monitoring and electronic word-of-mouth analysis, our system targets aspect-level organization and evidence-grounded summarization of reviews. Rather than predicting discrete emotions, we prioritize interpretable aspect clustering and sentiment aggregation to support user-facing restaurant understanding.

3 Description of the Method

We propose a **Hybrid Semantic-Syntactic Extraction Framework** that integrates dependency parsing with transformer-based embeddings. This architecture addresses the limitations of purely rule-based systems (low recall) and end-to-end deep learning (low explainability). The pipeline comprises five modules designed to extract fine-grained insights and synthesize objective summaries.

3.1 Data Preprocessing and Resource Initialization

We implement a context-aware normalization pipeline to ensure linguistic consistency while preserving domain-specific entities.

3.1.1 Constraint-Based Token Preservation

Standard tokenizers often fragment compound culinary terms (e.g., parsing *"Soft Shell Crab"* as *"Soft"* + *"Shell Crab"*). To resolve this, we employ a **Protected Phrase Injection** mechanism:

1. **Dynamic Menu Injection:** During data ingestion, the system parses establishment metadata (e.g., `dishes` field) to identify specific menu items.

2. **Token Locking:** Validated items (e.g., **"Soft Shell Crab"**) are registered as immutable tokens. This strictly prevents semantic fragmentation during downstream processing.

3.1.2 Context-Aware Filtering & Seed Initialization

To eliminate self-referential noise, we apply a **Dynamic Domain Stopword Filter** that adds the specific restaurant's name to the stopword list. For semantic alignment, we initialize **Aspect Seeds** (S_{Aspect}) stratified into **Positive** (S_{pos}) and **Negative** (S_{neg}) sub-groups.

- *Example (Service):*
  $S_{pos} = \{$"friendly", "attentive"$\}$,
  $S_{neg} = \{$"rude", "ignored"$\}$

- *Example (Taste):*
  $S_{pos} = \{$"delicious", "fresh"$\}$,
  $S_{neg} = \{$"bland", "stale"$\}$

This stratification enables context-dependent sentiment analysis (e.g., distinguishing *"slow"* as negative in Service but neutral/positive in *"slow-cooked"*).

3.2 Syntactic Dependency Phrase Extraction

To achieve high precision in boundary detection, we utilize **spaCy ('en_core_web_lg')** to traverse the dependency tree rather than using sliding windows. We apply four linguistic patterns:

1. **Recursive Modification:** Capturing nouns with their full adjective/adverb chains (e.g., *"surprisingly fresh sushi"*).

2. **Predicative Resolution:** Resolving `acomp`/`attr` dependencies to link adjectives in copular sentences back to subjects (e.g., *"The soup was salty"*).

3. **Verb-Argument Structure:** Extracting experiential verb-object pairs (e.g., *"loved the vibe"*).

4. **Domain Constraints:** Hard-coded rules for hygiene indicators (e.g., negation with *"gloves"*, *"masks"*).

All patterns incorporate **Negation Handling** by checking `neg` dependency tags to preserve semantic validity.

3.3 Two-Tier Semantic Aspect Classification

We employ a tiered strategy to balance efficiency and coverage:

1. **Tier 1 (Lexical Mapping):** Unambiguous tokens (e.g., *"expensive"* $\rightarrow$ *Price*) are immediately classified via a high-priority lexicon.

2. **Tier 2 (SBERT Embedding):** Unmatched phrases are encoded using **Sentence-BERT ('all-mpnet-base-v2')**. We calculate the maximum Cosine Similarity against aspect seeds:

$$Score(p, A) = \max_{s \in S_{Aspect}} \left(\frac{v_p \cdot s}{\|v_p\| \|s\|} \right) \quad (1)$$

Assignment occurs only if similarity exceeds strict thresholds ($\tau > 0.35$), effectively filtering irrelevant noise.

3.4 Vector-Space Sentiment Polarimetry

Unlike standard classifiers, we determine sentiment via **Vector-Space Polarimetry** relative to the aspect-specific seeds defined in Sec. 3.1.2. The polarity $Pol(p)$ is the differential similarity:

$$Pol(p) = \max_{s^+ \in S^A_{pos}} (\text{sim}(v_p, s^+)) - \max_{s^- \in S^A_{neg}} (\text{sim}(v_p, s^-)) \tag{2}$$

Phrases are labeled based on a margin $\delta = 0.05$. We augment this with **Contextual Valency Shifting** to detect intensifiers (e.g., *”too sweet”*) that invert standard polarities.

3.5 Ranking and Keyword Aggregation

To extract distinctive insights, we perform **Concept Normalization** (merging synonyms like *server/waiter*) while preserving Protected Phrases. Keywords are ranked using **TF-IDF**, where IDF penalizes generic descriptors (e.g., *”good food”*), prioritizing features unique to the specific establishment.

3.6 Aspect-Conditional Abstractive Summarization

To synthesize objective narratives from fragmented keywords, we deploy a **T5 (Text-to-Text Transfer Transformer)** model.

3.6.1 Model Configuration & Transfer Learning

We utilize the **T5-small** encoder-decoder architecture for its ability to generate coherent text conditioned on inputs. Due to data scarcity in local datasets, we apply **Cross-Domain Transfer Learning** by fine-tuning on the **Yelp Open Dataset**. Yelp's high lexical overlap with Google Maps ensures robust generalization.

3.6.2 Objective Perspective Transformation

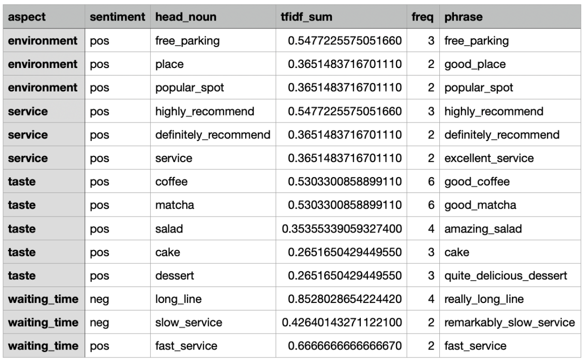
The model is trained with a **Seq2Seq objective**: inputting reviews concatenated with extracted keywords. We implement a **Perspective Transformation** by preprocessing training targets to replace first-person pronouns (e.g., *”I felt...”*) with third-person entities (e.g., *”Customers mentioned...”*). This compels the model to convert subjective user anecdotes into objective executive insights.

4 Experiments and Results

4.1 Pipeline Output

The proposed pipeline produces three structured outputs for each restaurant, providing a compact summary of user opinions extracted from Google Maps reviews.

1. **Aspect-Conditional Abstractive Summary:** Rather than performing simple sentence extraction, the system employs a fine-tuned T5 model to synthesize fragmented observations into a coherent, third-person narrative. This summary integrates distinct aspect-level signals to produce an objective executive overview (e.g., highlighting the contrast between *”inventive food”* and *”long wait times”*).
2. **Structured Aspect Profiling:** Review text is mapped onto a predefined taxonomy of five domains: *Taste*, *Service*, *Environment*, *Waiting Time*, and *Price*. As illustrated in **Figure 1**, the system extracts and ranks salient phrases for each domain based on their TF-IDF significance. These phrases are coupled with context-aware sentiment labels (“pos” or “neg”), providing a granular, interpretable decomposition of user feedback.
3. **Validated Dish Highlights:** Leveraging the Protected Phrase Injection mechanism, the system identifies signature menu items while filtering out generic terminology (e.g., ”food”, ”lunch”). This results in a curated list of specific entities (e.g., ”Soft Shell Crab”, ”Matcha Latte”) that appear frequently in positive contexts, distinguishing actual menu recommendations from general commentary.



aspect	sentiment	head_noun	tfidf_sum	freq	phrase
environment	pos	free_parking	0.5477225575051660	3	free_parking
environment	pos	place	0.3651483716701110	2	good_place
environment	pos	popular_spot	0.3651483716701110	2	popular_spot
service	pos	highly_recommend	0.5477225575051660	3	highly_recommend
service	pos	definitely_recommend	0.3651483716701110	2	definitely_recommend
service	pos	service	0.3651483716701110	2	excellent_service
taste	pos	coffee	0.5303300858899110	6	good_coffee
taste	pos	matcha	0.5303300858899110	6	good_matcha
taste	pos	salad	0.35355339059327400	4	amazing_salad
taste	pos	cake	0.2651650429449550	3	cake
taste	pos	dessert	0.2651650429449550	3	quite_delicious_dessert
waiting_time	neg	long_line	0.8528028654224420	4	really_long_line
waiting_time	neg	slow_service	0.42640143271122100	2	remarkably_slow_service
waiting_time	pos	fast_service	0.6666666666666670	2	fast_service

Figure 1: Step 1: Visualization of the aspect-level keywords extracted directly from raw reviews for Cafe Dulce. This represents the raw preprocessing output before API structuring.

Listing 1: Input Query JSON

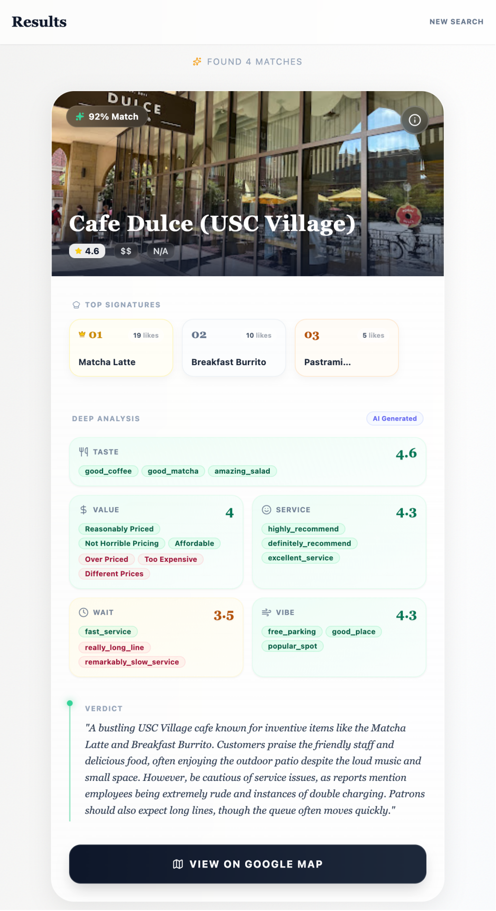
```
{
  "geolocation": { "lat": 34.0224, "lng"
      : -118.2851 },
  "search_radius_meters": 100,
  "preferences": {
    "nice_to_have": ["matcha"],
    "exclude_keywords": [""]
  }
}
```

Listing 2: Structured Output JSON (API Response)

```
{
  "id": 1,
  "name": "Cafe Dulce (USC Village)",
  "ai_summary": "A bustling cafe known
      for inventive items... Customers
      praise friendly staff but warn of
      long lines.",
  "aspects": {
    "taste": {
      "rating": 4.6,
      "summary": "Inventive & Delicious"
          ,
      "keywords": [
        { "phrase": "good_coffee", "
            sentiment": "pos" },
        { "phrase": "good_matcha", "
            sentiment": "pos" }
      ]
    },
    "waiting_time": {
      "rating": 3.5,
      "summary": "Long Lines",
      "keywords": [
        { "phrase": "long_line", "
            sentiment": "neg" }
      ]
    }
  }
}
```

Figure 2: Step 2: API Data Structure. The extracted insights shown in Figure 1 are packaged into a structured JSON response (Listing 2) based on the user query (Listing 1).

In addition, to address the limitations of standard platforms like Google Maps, where insights are often buried in unstructured text, we developed a web application designed to reverse-engineer the user discovery process. As shown in Figure 3, instead of forcing users to scroll through raw reviews, the interface visualizes our pipeline's structured output. This design bridges the gap between data availability and interpretability, allowing users to instantly assess a restaurant's quality through objective, evidence-based summaries.



"A bustling USC Village cafe known for inventive items like the Matcha Latte and Breakfast Burrito. Customers praise the friendly staff and delicious food, often enjoying the outdoor patio despite the loud music and small space. However, be cautious of service issues, as reports mention employees being extremely rude and instances of double charging. Patrons should also expect long lines, though the queue often moves quickly."

Figure 3: Step 3: The SpotLite web interface. This final application layer consumes the JSON data from Listing 2 to present a user-friendly restaurant profile.

4.2 LLM-based Evaluation (ChatGPT)

Instead of traditional metrics (accuracy, ROUGE, F1), we adopt a ChatGPT-based evaluation framework, motivated by recent work demonstrating that large language models can reliably serve as semantic evaluators for summarization and extraction tasks (Tam et al., 2023). We use ChatGPT as an external evaluator to assess the quality of our pipeline output.

A standardized evaluation prompt instructs the model to review:

- the raw input reviews,
- the pipeline's generated outputs (summary, aspect-level keywords, and dish extraction),
- and a structured evaluation rubric.

The rubric defines five dimensions, each scored on a 0–10 scale: *Aspect Accuracy*, *Keyword Quality*, *Coverage and Completeness*, *Noise and Irrelevance*, and *Dish Extraction Quality*. This setup

Table 1: LLM-based evaluation scores for Holbox

Dimension	Score
Aspect Accuracy	8.0
Keyword Quality	8.5
Coverage & Completeness	7.5
Noise & Irrelevance	7.0
Summary Quality	6.5

enables consistent quantitative scoring and detailed qualitative explanations for each dimension, while supporting scalable evaluation across a large number of restaurants.

The full evaluation prompt is provided in Appendix A for reproducibility.

4.3 Evaluation Results

We use a representative Holbox restaurant as an example. Table 1 reports LLM-based scores across five dimensions.

The pipeline performs strongly on *Aspect Accuracy* (8.0) and *Keyword Quality* (8.5), suggesting that aspect assignments and keywords align well with customer narratives. The introduction of TF-IDF-based phrase extraction contributes significantly to reducing generic or noisy keywords. It also achieves acceptable *Coverage and Completeness* (7.5) and *Noise and Irrelevance* (7.0), though PMI-based clustering occasionally introduces misgrouping errors. *Summary Quality* remains the main limitation (6.5): summaries are largely factual but often lack narrative richness and higher-level coherence. Overall, the results indicate robust aspect/keyword extraction while highlighting summarization as the primary bottleneck.

5 Discussion

The evaluation results highlight both the strengths and limitations of our pipeline and suggest several directions for further improvement.

5.1 Strengths

1. **High-precision aspect clustering:** Prototype-based clustering combined with keyword anchoring consistently grouped sentences into accurate aspects, improving interpretability over rule-based baselines.
2. **Strong keyword extraction quality:** TF–IDF with bigram and phrase mining produced specific, domain-relevant keywords. The LLM evaluator repeatedly noted their clarity and low noise.
3. **Effective noise reduction:** Preprocessing and phrase-filtering steps minimized irrelevant tokens and generic expressions, leading to cleaner outputs.

5.2 Current Limitations

1. **Limited global context in summaries:** While informative, extractive summaries do not capture broader narrative structure, emotional nuance, or contextual factors such as cleanliness, ambiance, or dining style.
2. **Coarse aspect taxonomy:** A fixed set of five aspects cannot fully represent cross-cutting topics (e.g., portion size, ambiance, value-for-money), leading to occasional misassignments. The trade-off between reproducibility (fixed taxonomy) and flexibility (dynamic topic discovery) remains unresolved.
3. **Keyword co-occurrence clustering errors:** PMI-based clustering occasionally merges weakly related bigrams due to sparse co-occurrence statistics.
4. **Subjectivity in LLM evaluation:** Different prompts or model versions may yield slightly different judgments.

6 Future Work

Building on the insights from the LLM-based evaluation, we outline several promising directions for further enhancing the pipeline:

1. **Improve summarization with aspect-aware Seq2Seq models:** Enhance summarization by fine-tuning T5 or Flan-T5 on aspect-tagged summaries rather than general Yelp-style data, improving narrative coherence and contextual depth.
2. **Expand aspect definitions or introduce dynamic aspect discovery:** Move beyond the five fixed aspects by automatically discovering aspect clusters or applying transformer-based topic modeling such as BERTopic (Grootendorst, 2022), enabling domain adaptation and finer granularity.
3. **Incorporate sentiment-weighted phrase extraction:** Weight extracted phrases by the sentiment intensity of the sentences in which they appear to better capture representative positive and negative cues.
4. **Stronger NER-based dish extraction:** Integrate a food-specific, fine-tuned NER model

to reduce false positives and ensure that extracted dish names correspond to real menu items.

5. **Adopt multi-LLM evaluation for greater reliability:** Use multiple LLM evaluators (e.g., GPT-4, Claude) to increase evaluation stability and reduce single-model bias.

7 Conclusion

This project demonstrates the feasibility of an end-to-end, non-LLM NLP pipeline that transforms unstructured restaurant reviews into structured, aspect-based insights. Our findings show that a combination of prototype-based clustering and keyword mining provides high-quality aspect summaries with minimal noise. An LLM-based evaluation framework enables scalable and quantitative assessment, revealing strengths in aspect accuracy and keyword clarity while highlighting opportunities for improvement in context-aware summarization and fine-grained aspect modeling. Future work will enhance summarization quality, expand aspect discovery, and adopt multi-model evaluation for greater reliability.

References

Paulo Carrasco and Sandra Dias. 2024. Enhancing restaurant management through aspect-based sentiment analysis and nlp techniques. *Procedia Computer Science*.

Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.

Jun Liu, Sike Hu, Fuad Mehraliyev, Haiyue Zhou, Yunyun Yu, and Luyu Yang. 2024. Recognizing emotions in restaurant online reviews: A hybrid model integrating deep learning and a sentiment lexicon. *International Journal of Contemporary Hospitality Management*, 36(9):2955–2976.

Idalia Maldonado Castillo, Ignacio Adrián Aguirre Miranda, and Alexis Olvera Mendoza. 2024. Automatic generation of restaurant reviews using natural language processing. In *Strategic Innovative Marketing and Tourism*, Springer Proceedings in Business and Economics. Springer.

Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. 2023. Evaluating the factual consistency of large language models through news summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5220–5255, Toronto, Canada. Association for Computational Linguistics.

Appendix A: ChatGPT Evaluation Prompt

You are an expert reviewer for an academic NLP project. I will give you:

1. A restaurant review dataset (raw text from `reviews.json`).

2. The output generated by my non-LLM NLP pipeline, including:
   - Summary for the restaurant,
   - Keywords for five aspects (Taste, Service, Environment, Waiting Time, Price),
   - Dish highlights extracted from reviews.

Your task is to evaluate the quality of my pipeline output.

Part 1: Input Data

[RAW REVIEWS]
[PIPELINE OUTPUT]

Part 2: Evaluation Task

Please evaluate the output based on the following five dimensions:

1. **Aspect Accuracy (0–10)**
   Are the aspect categories correctly summarized based on the reviews? Did the pipeline assign information to the correct aspects?

2. **Keyword Quality (0–10)**
   Are the extracted keywords relevant and meaningful? Do they reflect real customer concerns rather than generic or trivial words?

3. **Coverage & Completeness (0–10)**
   Does the summary capture the main ideas and recurring themes from the reviews? Are any important aspects or insights missing?

4. **Noise & Irrelevance (0–10)**
   How clean is the output? Are there useless, generic, or incorrect keywords? Is there topic pollution across aspects?

5. **Dish Extraction Quality (0–10)**
   Are the extracted dish names real dishes rather than generic food terms? Do they represent what customers frequently mentioned?