TrapCheck: AI-Powered Tourist Trap Detection

AI and Deep Learning Models Development: Course Project

Author: Omri Ashkenazi | Date: December 2025

1. Introduction & Motivation

Tourist traps represent a significant problem for travelers worldwide: establishments that exploit tourists through inflated prices, subpar quality, or deceptive practices. While online reviews provide valuable signals, the sheer volume of information and sophisticated manipulation tactics (fake reviews, review bombing) make manual assessment impractical.

TrapCheck addresses this challenge using a hybrid NLP architecture that combines deterministic pre-computed metrics with Large Language Model interpretation. Unlike pure LLM approaches that suffer from inconsistent analysis across runs, TrapCheck computes quantitative signals (reviewer credibility, keyword detection, date clustering) **before** the LLM processes any data. This ensures consistent, reproducible assessments while leveraging the LLM's natural language capabilities for interpretation and explanation.

The system is **venue-agnostic**, supporting multiple venue types including restaurants, cafés, bars, museums, attractions, tours, shops, and markets, making it a general-purpose solution for travel decision-making. Automatic venue type detection adjusts keywords, signals, and RAG retrieval for optimal analysis per category.

2. Dataset and Preprocessing

2.1 RAG Knowledge Base

The Retrieval-Augmented Generation (RAG) database contains **149 curated global venues** classified as Tourist Traps (score 60-95), Local Gems (score 5-35), or Mixed (score 35-65). The database was created using **Gemini 3 Pro** with extensive web research, collecting data from travel forums (TripAdvisor, Lonely Planet), Reddit communities (r/travel, r/food, city subreddits), food blogs, and local review sites. Each entry includes venue name, location, category, verdict, trap score, and descriptive keywords for retrieval.

Category	Count	Percentage
Restaurants	55	37%
Attractions	39	26%
Cafés	15	10%
Street Food	13	9%
Markets	13	9%

Bars	11	7%
Tours	3	2%

Table 1: RAG Database Category Distribution

2.2 Evaluation Datasets

Two evaluation frameworks were developed during the project. The initial **v1 evaluation** used mock review data for 4 venues with manually assigned ground truth scores, enabling controlled experiments without API costs. The comprehensive **v2 evaluation** uses 30 stratified venues sampled from the RAG database (10 traps, 10 gems, 10 mixed), with 3 runs per venue (90 total runs) and leave-one-out evaluation to prevent data leakage. Synthetic reviews include realistic noise (1-2 contradictory reviews, ambiguous 3-star reviews) to prevent overfitting to clean patterns.

Venue	Location	Category	Ground Truth
Pizzeria Da Michele	Naples	Authentic	25
Olive Garden Times Sq.	NYC	Tourist Trap	85
Carlo Menta	Rome	Mixed	50
Katz's Delicatessen	NYC	Mixed	35

Table 2: v1 Mock Evaluation Venues (4 venues)

3. Model Design: RAG-Enhanced Analysis Pipeline

TrapCheck employs a three-stage architecture that separates deterministic signal extraction from LLM interpretation, ensuring reproducibility while leveraging language model capabilities.

3.1 Pre-computed Metrics Layer

The metrics layer (*src/metrics.py*) extracts quantitative signals deterministically before LLM analysis, ensuring 100% signal stability across runs:

- **Reviewer Credibility Scoring:** Based on review count, Local Guide status, and photo contributions. Separate averages for positive vs. negative reviewers detect 'credibility inversion' (when negative reviewers are more credible).
- **Keyword Detection:** Identifies trap awareness terms ('tourist trap', 'scam', 'rip off'), manipulation accusations ('fake review', 'paid'), and venue-specific quality complaints adapted per venue type.
- **Date Clustering Analysis:** Detects suspicious patterns of positive reviews on same days, indicating potential review manipulation.
- **Language Analysis:** Uses langdetect to compare sentiment between tourist and local language reviews, detecting credibility splits by language.
- **External Signals:** Web search via Gemini 2.0 Flash Lite with Google Search grounding retrieves opinions from Reddit, TripAdvisor forums, and food blogs, plus tourist hotspot proximity detection.

3.2 RAG Calibration Pipeline

Two retrieval approaches were implemented and compared for score calibration:

- **Keyword RAG** (production default): Jaccard-like keyword overlap scoring with no external dependencies. Retrieves 6 balanced examples (2 traps, 2 gems, 2 mixed) per query. ~0.1s latency overhead.
- **Vector RAG** (alternative): ChromaDB vector store with all-MiniLM-L6-v2 embeddings using cosine similarity search. Better for larger databases (500+ examples). ~0.9s latency overhead.

3.3 LLM Integration

The final stage uses **Gemini 2.5 Flash Lite** (selected for speed and cost optimization after initial development with Gemini 2.5 Flash) with structured JSON output schema to interpret metrics and generate verdicts. The JSON schema constraints ensure consistent output format, effectively eliminating temperature-based variance. RAG examples provide few-shot context for score calibration. Reviews sent to the LLM include both credible negative reviews (5) and credible positive reviews (3) to avoid confirmation bias.

4. Experimental Process and Methodology

Development followed an iterative approach, with experiments designed to address specific challenges encountered during testing:

- **Phase 1 - MVP:** Initial implementation with Gemini 2.5 Flash analyzing Google reviews. Significant inconsistency observed (20+ point variance on identical data), which could skew verdicts from 'gem' to 'possible trap'.
- **Phase 2 - Stability Improvements:** Added web search, language analysis, and location/tourist hotspot detection. Improved consistency substantially but room for improvement remained.
- **Phase 3 - Model Optimization:** Downgraded to Gemini 2.5 Flash Lite for faster, cheaper inference with comparable quality.
- **Phase 4 - RAG Integration:** Created 149-venue RAG database with Gemini 3 Pro assistance. Implemented both keyword and vector retrieval methods.
- **Phase 5 - Comprehensive Evaluation:** Designed v2 evaluation framework with 30 stratified venues, 3 runs each (90 total), measuring category accuracy, score MAE, and variance.

Evaluation metrics included **Category Accuracy** (correct verdict classification), **Score Accuracy** (within ±15 of ground truth), **Mean Absolute Error** (MAE), and **Standard Deviation** across runs. Temperature studies (0.0, 0.5, 1.0, default) were also conducted.

5. Results

5.1 Configuration Comparison (v2 Evaluation)

The v2 evaluation framework tested multiple configurations across 30 stratified venues (90 total runs per experiment):

Configuration	Category Acc.	Within ±15	MAE	StdDev
Baseline (no RAG)	90.0%	77.8%	13.6	3.27
temp_0.0	90.0%	84.4%	12.7	1.64
temp_0.5	90.0%	81.1%	13.0	2.19
RAG Keyword	95.6%	93.3%	9.3	1.38
RAG Vector	94.4%	93.3%	9.9	2.06

Table 3: v2 Evaluation Results (30 venues × 3 runs = 90 total runs per configuration)

Key Finding: RAG Keyword achieved a 32% reduction in MAE (13.6 → 9.3) and 95.6% category accuracy with minimal latency overhead (+3%), outperforming both baseline and Vector RAG approaches.

5.2 Per-Category Analysis

The most significant improvement occurred in the 'mixed' category, where the model previously struggled most:

Category	Baseline	RAG Keyword	Improvement
Tourist Trap	100%	97%	-3% (minor)
Local Gem	100%	100%	0%
Mixed	70%	90%	+20%

Table 4: Per-Category Accuracy Comparison

The baseline already achieved 100% accuracy on clear-cut tourist traps and local gems, demonstrating that the pre-computed metrics layer effectively identifies strong signals. RAG calibration provided crucial context for ambiguous 'mixed' venues where explicit 'tourist trap' keywords appeared alongside genuine positive signals.

5.3 Key Findings

- **Pre-computed metrics ensure 100% signal stability** across runs, validating the hybrid architecture approach. The same venue analyzed multiple times always produces identical signal detection.
- **Keyword RAG outperformed Vector RAG** on this 149-example database (95.6% vs 94.4% accuracy), suggesting small curated databases work well with keyword matching.

• **Temperature has minimal effect** due to JSON schema output constraints. All temperature values achieved 90% accuracy; temp_0.0 only reduces variance slightly.

• **Mixed venues dramatically improved:** Category accuracy jumped from 70% to 90% with RAG, the largest single improvement.

• **Variance correlates with signal ambiguity,** not inherent LLM randomness. Clear cases show 3-4 StdDev vs. 5+ for ambiguous venues.

6. Analysis and Limitations

6.1 Architectural Insights

The hybrid architecture's success stems from a clear separation of concerns: deterministic metrics handle quantifiable signals (credibility scores, keyword counts, date patterns) while the LLM focuses on interpretation and explanation. This prevents the LLM from being influenced by emotional review language and ensures auditable signal detection. The pre-computed approach also reduces token usage by summarizing patterns rather than passing raw review text.

6.2 Limitations

• **Mixed Venue Calibration:** System occasionally over-classifies ambiguous venues as traps when explicit 'tourist trap' keywords appear, even if overall sentiment is mixed.

• **Synthetic Evaluation:** While v2 uses 30 venues with realistic noise, results are on synthetic reviews generated from RAG entries; real-world performance with actual API data may vary.

• **English-Centric:** Best performance on English reviews. Language detection helps identify tourist vs. local patterns but coverage for non-English sources varies.

• **RAG Database Size:** 149 examples may not cover all venue types equally. Tours are underrepresented (only 3 examples, 2%).

• **Persistent Edge Cases:** Some venues (e.g., Harry's Bar Venice) are consistently misclassified across all experiments, suggesting certain venue profiles remain challenging.

7. Conclusion and Future Work

TrapCheck demonstrates that a hybrid architecture combining deterministic pre-computed metrics with LLM interpretation achieves consistent, reproducible venue assessment. The RAG-enhanced pipeline delivers **32% improvement in score accuracy** (MAE 13.6 → 9.3) and **95.6% category accuracy** with minimal latency overhead. The most significant impact was on 'mixed' venues, where accuracy improved from 70% to 90%.

The key architectural insight: computing signals before LLM analysis ensures quantitative reliability (100% signal stability) while leveraging natural language capabilities for interpretation. Keyword-based RAG proved sufficient for a curated database of 149 examples, outperforming vector embeddings while requiring no external dependencies.

Repository: github.com/ashkenazzio/trap-check | **Course:** AI and Deep Learning Models Development, December 2025