Title Page

Decoding Diner Safety: Summarizing Chicago's Restaurant Health Violations

Carolina Aldana Yabur || Shlok Nandkishor Goud || Zahid Rahman

(1) Executive Summary

📌 **Problem**

- Chicago's food inspection data is high-quality but hard to interpret in raw table format

📊 **Data Source**

- Chicago Food Inspections dataset (official public API, minimal cleaning).
- Clean violation text ideal for NLP
- Real-time access for live queries

🧠 NLP Analysis

- Fine-tuned OpenAI GPT (LoRA-adapted flan-t5-small) on enriched JSONL
- Generated summaries, keyword tags, and soft safety verdicts
- Achieved modest ROUGE gains and Low BERTScore

🛠️ Technical Implementation

- Backend (Flask API): Real-time API pipeline successfully built
- Frontend (React + Axios): Clean, responsive UI ready for deployment
- ✅ Integrations completed

🔍 Key Findings

- Many outputs were blank, repetitive, or incoherent
- Limited data affected generalization

(2) Problem Definition & Significance

The target clients for this project are health-conscious consumers in Chicago who want quick and understandable insights into the sanitary conditions of local restaurants. Currently, the Chicago Data Portal provides comprehensive health inspection data in raw tabular form, but this format is not user-friendly or accessible for the general public. As a result, most consumers are unaware of violations that may impact their health.

This lack of transparency presents a public health concern. According to the CDC, approximately 1 in 6 Americans (48 million people) get sick from foodborne illnesses each year. In Chicago alone, hundreds of restaurants receive citations annually, yet there is no mainstream tool that translates this data into consumer-friendly insights.

Our goal is to create a web application that simplifies health inspection reports by summarizing violations using natural language processing, extracting key issues, and offering a soft verdict on safety. This tool will empower users to make safer dining choices and foster greater public accountability in food service establishments.

(3) Prior Literature

- Wang et al. (2017) used Yelp reviews to predict foodborne illness with SVM and RNN models. Focused on detection, not consumer usability.
- Dumne et al. (2020) applied TextRank for extractive summarization—helpful for condensing lengthy text but not tailored to inspections.
- Kannan et al. (2019) forecasted critical violations using inspection metadata; aimed at inspector efficiency, not public awareness.
- Mejia et al. (2019) built hygiene word dictionaries from crowd-labeled Yelp reviews to track post-inspection decline—data-rich but not user-facing.
- Kang et al. (2013) combined sentiment and metadata to predict inspection scores; strong accuracy but lacked summarization.
- Mahmood & Khan (2019) used ML on inspection data for classification but did not explore NLP.
- Schomberg et al. (2016) fused social media with inspection records for early outbreak warnings—good integration but not aimed at public summaries.

Our Contribution:
We plan to go beyond prediction by using NLP to summarize violations, extract keywords, and deliver a consumer-facing safety verdict, improving public accessibility.

(4) Data Source & Preparation

- **Source**: Chicago Food Inspections from the official Chicago Data Portal — a clean, government-maintained dataset with no major cleaning required.
- **Advantages**: High data quality, detailed violation text ideal for NLP, and a public API enabling real-time integration into web applications.
- **Note**: While the API was a key strength, integration challenges were more greater than expected and significantly slowed progress towards a polished product.

What's in this Dataset?

Rows	Columns	Each row is a	Row Identifier
291K	17	Food Inspection	Inspection ID

Columns (17)

Column Name	Description	API Field Name	Data Type
Inspection ID		inspection_id	Number
DBA Name	Doing Business As	dba_name	Text
AKA Name	Also Known As	aka_name	Text
License #		license_	Number
Facility Type		facility_type	Text
Risk		risk	Text
Address		address	Text
City		city	Text
State		state	Text
Zip		zip	Number
Inspection Date		inspection_date	Floating Timestamp
Inspection Type		inspection_type	Text
Results		results	Text
Violations		violations	Text
Latitude		latitude	Number

Rows per page: 15 1-15 of 17

(5) Methodology: Analytical Approach & Rationale

NLP Approach (Zahid):

- Generated summaries, keywords, and safety verdicts using OpenAI GPT
- Fine-tuned a quantized, LoRA-adapted flan-t5-small transformer on enriched JSONL data
- Chosen for its customizability and lightweight deployment — but results were weaker than expected
- In hindsight, model underperformance may be due to limited training data, as later hinted at by faculty advice to use ChromaDB + LLM + reranking

Backend Integration (Shlok):

- Developed API to connect user queries → Chicago Data Portal → model inference
- Intended to enable real-time responses, but integration failed due to mismatched output formats and model inconsistency

Frontend Design (Carolina):

- Built a clean, responsive UI to display summaries and keywords alongside official inspection data
- Consulted external faculty and recommended switching to embedding-based retrieval, but feedback came too late to pivot

----

Why These Approaches?

We aimed for a modular, cost-effective NLP pipeline using fine-tuned transformers and real-time data — balancing performance, flexibility, and usability.

Outcome:

All components were independently functional, but integration stalled. Shlok's sudden hospitalization during the final stretch ultimately hampered end-to-end delivery. However, we were able to get a rough POC ready.

(6) Experiments & Results

Epoch	Training Loss	Evaluation Loss
1	3.31	2.83
2	2.81	2.29
3	2.65	2.16

Model	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
Base	0.129	0.032	0.101	0.465
Fine-Tuned	0.155	0.037	0.122	0.435

- Observations:
  - Fine-tuned model showed small ROUGE improvements but underperformed on BERTScore
  - Many outputs were blank, repetitive, or gibberish due to mismatched prompts at inference
  - Model was trained to emit structured JSON but was evaluated using plain-text prompts
  - Evaluation metrics were impacted by invalid outputs, not just model quality
  - Faculty later suggested training data was too limited for strong generalization

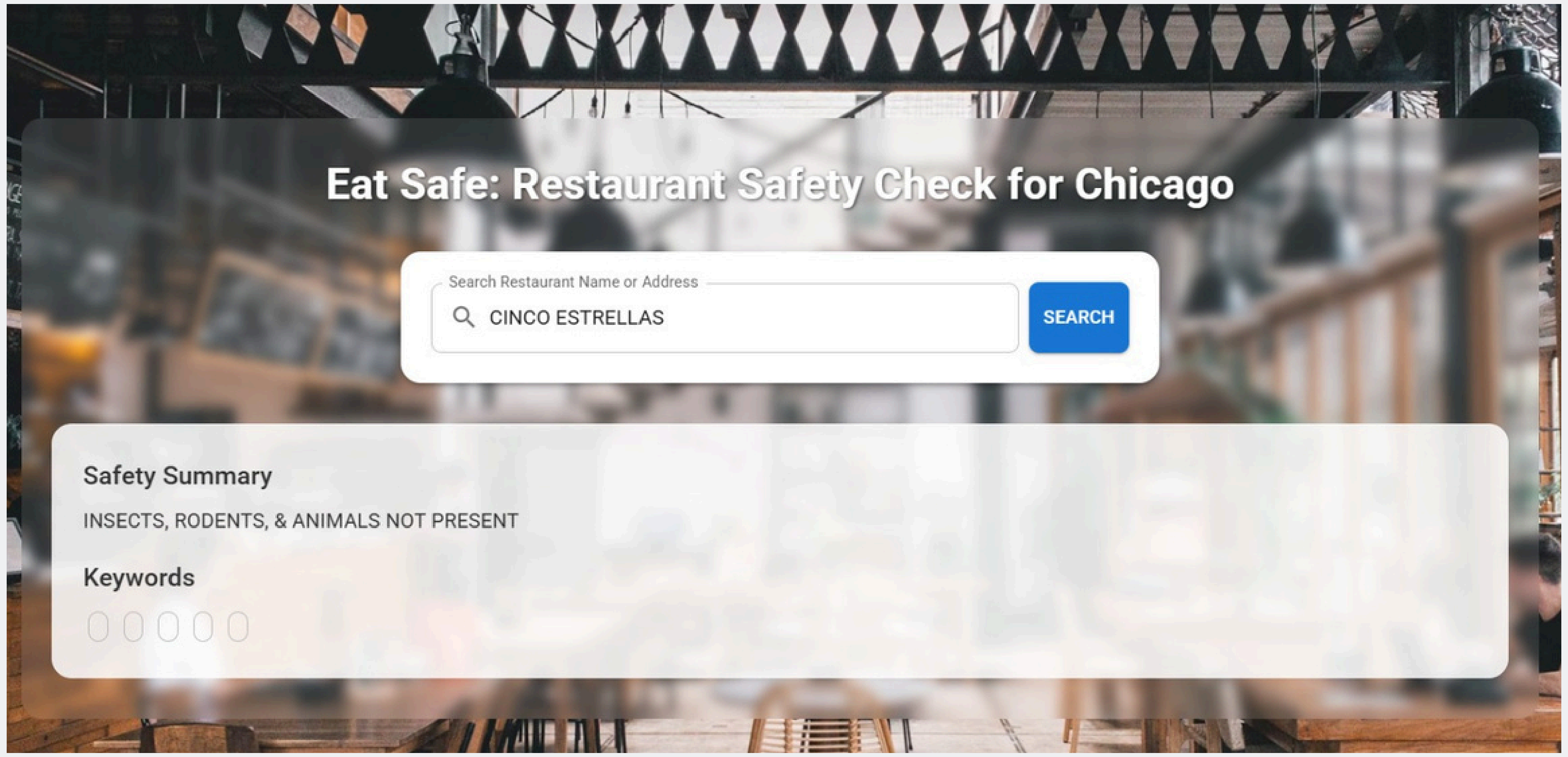
- Takeaway:

While the transformer trained successfully and improved slightly in ROUGE, the evaluation pipeline didn’t align with the training format—leading to deceptively low scores. With better prompt consistency and more training data, performance could likely improve. More advanced pipelines (e.g. ChromaDB + LLM + reranking) were recommended but came too late to implement.

Eat Safe: Restaurant Safety Check for Chicago

What it does

Transforms Chicago health inspection records into a concise safety summary and the top 5 keywords using a fine-tuned FLAN-T5 model.



Architecture Overview

🧠 Backend (Flask API)

- Connects to Chicago's open data portal.
- Handles POST requests (/api/restaurant-info) with a search term.
- Returns a one-line AI-generated summary and keyword set.
- Modular services for API fetch, model inference, and response formatting.

💻 Frontend (React + Axios)

- Clean Material UI search interface built with React 18 and Axios.
- User enters a restaurant name or address in the search field; fetchSafetyData(query) sends a POST request to the Flask API.
- On success, the page displays a safety summary along with keywords.

(7) Insights & Future Work

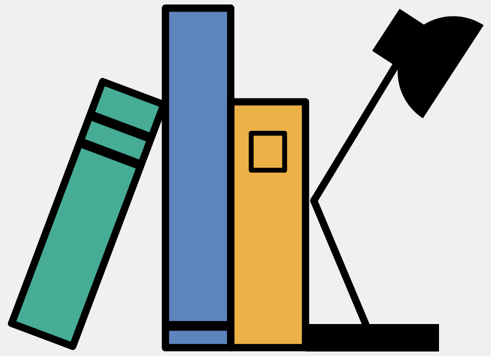
📌 Key Recommendations

- Upgrade the Model
  - Replace flan-t5-small with a more capable LLM (e.g., GPT-3.5, Mistral-7B)
  - ✅ Action: Choose a stronger base model for improved output quality
- Fix Adapter Loading Issue
  - Warning “Adapter weights not found” indicates model fallback to base
  - ✅ Action: Ensure proper adapter path and loading in deployment pipeline
- Use More Training Data
  - Only a fraction of ~290,000 records were used for fine-tuning
  - ✅ Action: Expand dataset and use data augmentation or reranking

🚀 Longer-Term Extensions

- Multilingual interface: Serve Chicago’s Spanish- and Polish-speaking communities.

Sources



(8) References

- Dumne, R., Gavankar, N. L., Bokare, M. M., & Waghmare, V. N. (2024). Automatic Text Summarization using Text Rank Algorithm. 2024 3rd International Conference for Advancement in Technology (ICONAT), 1–6. https://doi.org/10.1109/ICONAT61936.2024.10775241
- Kang, J. S., Kuznetsova, P., Luca, M., & Choi, Y. (2013). Where not to eat? Improving public policy by predicting hygiene inspections using online reviews. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1443-1448).
- Kannan, V., Shapiro, M. A., & Bilgic, M. (2019). Hindsight analysis of the Chicago food inspection forecasting model. arXiv.org.
- Mahmood, A., & Khan, H. U. (2019). Identification of critical factors for assessing the quality of restaurants using data mining approaches. Electronic Library, 37(6), 952–969. https://doi.org/10.1108/EL-12-2018-0241
- Mejia, J., Mankad, S., & Gopal, A. (2019). A for effort? Using the crowd to identify moral hazard in New York City restaurant hygiene inspections. Information Systems Research, 30(4), 1363–1386. https://doi.org/10.1287/isre.2019.0866
- Schomberg, J. P., Haimson, O. L., Hayes, G. R., & Anton-Culver, H. (2016). Supplementing public health inspection via social media. PLOS One, 11(3), e0152117–e0152117. https://doi.org/10.1371/journal.pone.0152117
- Wang, Z., Balasubramani, B. & Cruz, I (2017). Predictive analytics using text classification for restaurant inspections. Proceedings of the 3rd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics. https://doi.org/10.1145/3152178.3152192