School of Engineering, University of Guelph

Winter 2026

Food Hazard Project

Dr. Fattane Zarrinkalam

Alireza Naseri (Master's)

Date: February 28, 2026

Contents

1 Introduction 3
1.1 Project Goal and Scope . . . 3
1.2 Contributions . . . 3
2 Dataset 4
2.1 Dataset Source . . . 4
2.2 Dataset Schema . . . 4
2.3 Label Space . . . 5
2.4 Class Imbalance . . . 5
2.4.1 Hazard Categories (Top-5) . . . 5
2.4.2 Product Categories (Top-5) . . . 6
3 Methods 6
3.1 Prompt-Based Inference . . . 6
3.1.1 Zero-Shot Strict Prompting . . . 6
3.1.2 Results (Strict) . . . 7
3.1.3 Zero-Shot Grounded Prompting with Canonicalization . . . 9
3.1.4 Results (Grounded + Canon) . . . 9
3.1.5 Analysis and Error Patterns . . . 10
3.1.6 Few-Shot Prompting (Targeted 4-shot) . . . 11
3.1.7 Results (Few-shot 4-shot) . . . 11
3.1.8 Analysis . . . 12
3.2 Supervised Fine-Tuning (SFT) . . . 13
3.2.1 Multi-task fine-tuning setup . . . 13
3.2.2 Multi-task results and diagnosis . . . 14
3.2.3 Product-category single-task fine-tuning . . . 14
3.2.4 Summary of key findings . . . 14
3.3 Retrieval-Augmented Generation (RAG) . . . 15
3.3.1 Knowledge Base Construction . . . 15
3.3.2 Embedding and Retrieval Setup . . . 15
3.3.3 Validation Results and Ablation on k . . . 15
3.3.4 Test Results . . . 16
3.3.5 Analysis . . . 16
3.3.6 Hybrid RAG: Task-Specific Retrieval Design . . . 17
4 Evaluation and Comparative Analysis 18
4.1 4.1 Quantitative Comparison Across Paradigms . . . 18
4.2 Hazard vs. Product: Asymmetric Task Behavior . . . 19
4.3 Rare-Class and Long-Tail Behavior . . . 19

4.4 Structured Error Analysis 19
4.5 Computational Trade-offs 20
5 Discussion 20
5.1 When Is Prompting Enough? 20
5.2 Why Supervised Fine-Tuning Outperformed Retrieval 21
5.3 Why RAG Did Not Improve Hazard Performance 21
5.4 Safety-Critical Implications 21
5.5 Practical Constraints and Engineering Trade-offs 22
5.6 Key Takeaways 22
6 Conclusion 23
A Detailed Experimental Results 24
B Reproducibility and Configuration Details 24

1 Introduction

Food hazard detection from unstructured web reports is a safety-critical NLP task with direct real-world consequences. Automated systems that monitor public sources (e.g., official recall websites, news portals, and social media) can assist experts by rapidly flagging incidents and categorizing both the underlying *hazard* and the affected *food product*. However, such systems must be reliable and transparent: incorrect predictions may lead to missed recalls or unnecessary economic damage, while unstable model behavior can undermine trust in automated monitoring pipelines.

This project is based on the dataset and task formulation introduced in *SemEval-2025 Task 9: The Food Hazard Detection Challenge*, where food-incident reports are annotated under a long-tail class distribution. The shared task defines two subtasks: a coarse-grained category prediction task (ST1) and a more fine-grained "vector" prediction task (ST2). In ST1, models predict one of **10 hazard categories** (e.g., *allergens*, *biological*, *foreign bodies*) and one of **22 product categories** (e.g., *meat, egg and dairy products*, *cereals and bakery products*). The challenge highlights that long-tail imbalance and semantic overlap between classes make this problem non-trivial, and that robustness and interpretability are essential due to the high-stakes nature of food safety applications.

1.1 Project Goal and Scope

The goal of this project is to develop practical experience in applying *Large Language Models (LLMs)* to food hazard detection and to systematically compare three LLM paradigms under a consistent experimental setup:

1. **Prompt-Based Inference:** zero-shot and few-shot prompting using an instruction-tuned LLM without updating parameters, emphasizing output stability, interpretability, and reliability.

2. **Supervised Fine-Tuning (SFT):** fine-tuning a pretrained transformer classifier on the labeled training data to quantify the benefits of task-specific adaptation, particularly for rare categories.

3. **Retrieval-Augmented Generation (RAG):** running an end-to-end RAG pipeline (without implementing retrieval/indexing from scratch) and evaluating how external knowledge and retrieved context affect accuracy, robustness, and long-tail behavior.

Consistent with the assignment requirements, all approaches are evaluated using standard classification metrics (Accuracy, Micro-F1, Macro-F1, and per-class F1), with particular attention to rare hazard categories. In addition, we conduct a structured error analysis to identify common failure modes such as confusion between semantically similar hazards, ambiguous incident descriptions, missing context, and retrieval-induced noise.

1.2 Contributions

This report makes the following contributions:

- A concise dataset familiarization summary, including schema, label spaces, and class imbalance observations (top-frequency labels and long-tail behavior).
- Reproducible implementations of three LLM paradigms (prompting, SFT, and RAG) for ST1 food hazard-category and product-category prediction.
- A consistent evaluation protocol enabling a fair comparison of effectiveness, stability, and computational trade-offs across paradigms.
- A structured failure analysis that surfaces practical error patterns and limitations relevant to safety-critical deployment scenarios.

2 Dataset

2.1 Dataset Source

We use the dataset released for *SemEval-2025 Task 9: The Food Hazard Detection Challenge*. The dataset contains manually annotated food recall reports collected from official food safety agencies. Each instance consists of structured metadata fields, textual content, and both coarse-grained (category-level) and fine-grained (vector-level) labels.

The official data split includes:

- **Train:** 5,082 samples
- **Validation:** 565 samples
- **Test:** 997 samples

In total, the dataset contains **6,644** food-incident reports.

2.2 Dataset Schema

Split	#Rows	#Cols	Columns
Train	5082	11	year, month, day, country, title, text, hazard-category, product-category, hazard, product
Valid	565	11	Same as train
Test	997	11	Same as train

Table 1: *Dataset schema for SemEval-2025 Task 9. All splits share identical structure.*

Each record contains temporal metadata (`year, month, day`), geographic information (`country`), textual inputs (`title` and full `text`), and four label columns: two category-level labels (`hazard-category`, `product-category`) and two fine-grained labels (`hazard`, `product`).

2.3 Label Space

Split	Label Column	#Unique Values
Train	hazard-category	10
Train	product-category	22
Train	hazard	128
Train	product	1022
Valid	hazard-category	9
Valid	product-category	18
Valid	hazard	93
Valid	product	312
Test	hazard-category	10
Test	product-category	20
Test	hazard	110
Test	product	447

Table 2: *Number of unique label values per split.*

For ST1 (category-level prediction), the label space consists of **10 hazard categories** and **22 product categories**. For ST2 (fine-grained prediction), the number of unique hazard and product labels exceeds one hundred, resulting in a substantially more challenging long-tail classification problem.

2.4 Class Imbalance

2.4.1 Hazard Categories (Top-5)

Split	Top-5 Hazard Categories (count)
Train	allergens (1854), biological (1741), foreign bodies (561), fraud (371), chemical (287)
Valid	allergens (207), biological (194), foreign bodies (63), fraud (41), chemical (28)
Test	allergens (365), biological (343), foreign bodies (111), fraud (75), chemical (52)

Table 3: *Top-5 most frequent hazard categories.*

2.4.2 Product Categories (Top-5)

Split	Top-5 Product Categories (count)
Train	meat, egg and dairy products (1434), cereals and bakery products (671), fruits and vegetables (535), prepared dishes and snacks (469), seafood (268)
Valid	meat, egg and dairy products (146), cereals and bakery products (75), prepared dishes and snacks (56), fruits and vegetables (52), soups, broths, sauces and condiments (36)
Test	meat, egg and dairy products (282), cereals and bakery products (121), fruits and vegetables (103), prepared dishes and snacks (92), seafood (60)

Table 4: *Top-5 most frequent product categories.*

The dataset exhibits a pronounced long-tail distribution. Across all splits, the hazard categories *allergens* and *biological* dominate the label space, while several categories (e.g., migration, food additives and flavourings) are significantly underrepresented.

A similar imbalance is observed for product categories, where *meat, egg and dairy products* is consistently the most frequent class.

This imbalance has important methodological implications: macro-F1 and per-class F1 scores are necessary to avoid inflated performance due to dominant classes, and rare categories are expected to be particularly challenging for both prompt-based and fine-tuned LLM approaches.

3 Methods

3.1 Prompt-Based Inference

3.1.1 Zero-Shot Strict Prompting

In the zero-shot strict setting, we evaluate whether an instruction-tuned Large Language Model (LLM) can perform food hazard and product classification without task-specific fine-tuning. We use `llama3.1:8b` and constrain its output using a carefully engineered prompt that explicitly restricts predictions to predefined label sets.

Prompt Design. The prompt includes:

- Explicit definitions for all 10 hazard categories,
- A complete list of 22 allowed product categories,
- Clear formatting constraints requiring exactly two output lines,

- A strict instruction prohibiting invention of unseen labels.

The model is required to output:

```
hazard-category: <allowed_label>
product-category: <allowed_label>
```

Any output not matching the allowed label space is automatically rejected during parsing.

Output Validation. Predictions are post-processed to ensure:

- The predicted labels exist in the allowed label lists,
- Formatting follows the required two-line structure,
- Invalid outputs are marked as parsing failures.

3.1.2 Results (Strict)

Validation Set.

- Parse OK rate: 0.993
- Average latency: 0.63 seconds/sample

Hazard-category performance:

- Accuracy: 0.877
- Macro-F1: 0.588
- Weighted-F1: 0.860

Product-category performance:

- Accuracy: 0.390
- Macro-F1: 0.383
- Weighted-F1: 0.391

Test Set.

- Parse OK rate: 0.991
- Average latency: 0.68 seconds/sample

Hazard-category performance:

- Accuracy: 0.887
- Macro-F1: 0.598

- Weighted-F1: 0.874

Product-category performance:

- Accuracy: 0.376
- Macro-F1: 0.388
- Weighted-F1: 0.377

1. Stability and Format Reliability. The strict prompting strategy achieved a very high parse success rate (>99%), demonstrating that instruction-following LLMs can reliably adhere to structured output constraints when explicitly guided.

2. Hazard Classification Behavior. Hazard-category prediction shows strong overall accuracy (0.88), largely driven by dominant classes such as *allergens* and *biological*. However, macro-F1 (0.59) is substantially lower than accuracy, reflecting severe long-tail effects.

Rare categories such as *migration*, *food additives and flavourings*, and *other hazard* receive very low F1 scores. In several cases, the model fails to predict these categories at all, indicating a bias toward frequent hazard types.

A notable failure mode appears in the *fraud* category, where recall is particularly low. This suggests the model struggles with identifying intentional deception when explicit contamination cues are absent.

3. Product Classification Difficulty. Product-category performance is significantly weaker (accuracy 0.38). Unlike hazard detection, product classification requires more fine-grained semantic reasoning about the recalled item itself rather than the hazard source.

The model exhibits strong overprediction of dominant categories such as *meat, egg and dairy products* and *prepared dishes and snacks*, leading to inflated recall for those classes but poor precision elsewhere.

Several classes receive near-zero F1 (e.g., *soups, broths, sauces and condiments*), highlighting difficulty in distinguishing nuanced food types without task-specific adaptation.

4. Long-Tail Sensitivity. The gap between weighted-F1 and macro-F1 confirms a heavy class imbalance effect. Zero-shot prompting primarily captures high-frequency patterns but fails to generalize reliably to rare or underrepresented categories.

5. Practical Trade-Off. Zero-shot strict prompting offers:

- Extremely low engineering overhead,
- Strong formatting reliability,
- Competitive hazard detection performance.

However, it suffers from:

- Severe long-tail weakness,
- Poor fine-grained product classification,
- Bias toward dominant hazard types.

These findings motivate exploring task-specific adaptation (Section 3.2) and knowledge augmentation (Section 3.3).

3.1.3 Zero-Shot Grounded Prompting with Canonicalization

Why moving from strict to grounded+canonicalization? Although the strict zero-shot setup enforces a closed label space, we observed that instruction-tuned LLMs may still output *semantically correct* labels with minor surface-form variations (e.g., American vs. British spelling such as *flavorings* vs. *flavourings*, punctuation differences, or singular/plural forms like *foreign body* vs. *foreign bodies*). In a safety-critical setting, these formatting-level deviations should not be conflated with genuine classification errors. Therefore, we moved to a grounded prompting setup with a **soft canonicalization layer** that maps common variants to the official label strings before enforcing membership in the allowed label lists. This change improves evaluation stability and reduces artificial errors caused purely by output normalization, without relaxing the closed-world assumption of the task.

Method. We keep the same LLM (`llama3.1:8b`) and the same strict output format (two-line output). The only modification is the parsing stage:

- Extract predicted labels from the two required lines,
- Normalize whitespace/punctuation,
- Apply small canonical maps (`HAZ_CANON`, `PROD_CANON`) to convert frequent surface variants into the exact official labels,
- Accept predictions only if they match the fixed label spaces after canonicalization.

3.1.4 Results (Grounded + Canon)

Validation Set.

- Parse OK rate: 0.993
- Avg latency: 0.64 seconds/sample

Hazard-category:

- Accuracy: 0.877
- Macro-F1: 0.574
- Weighted-F1: 0.859

Product-category:

- Accuracy: 0.406
- Macro-F1: 0.392
- Weighted-F1: 0.409

Test Set.

- Parse OK rate: 0.989
- Avg latency: 0.66 seconds/sample

Hazard-category:

- Accuracy: 0.884
- Macro-F1: 0.562
- Weighted-F1: 0.873

Product-category:

- Accuracy: 0.401
- Macro-F1: 0.367
- Weighted-F1: 0.409

3.1.5 Analysis and Error Patterns

Effect of canonicalization. On the validation set, hazard-category macro-F1 decreased slightly from 0.588 (strict) to 0.574 (canon), while product-category accuracy improved from 0.390 to 0.406. Since canonicalization primarily targets formatting variants, large metric shifts are not expected; instead, the main benefit is **evaluation robustness**—ensuring that equivalent labels are not counted as errors due to spelling or punctuation.

Hazard-category behavior. Performance remains strong for high-support classes (*allergens*, *biological*, *foreign bodies*), while long-tail categories continue to dominate macro-F1 limitations. In particular, *fraud* shows low recall (e.g., 0.15 on validation and 0.35 on test), indicating that the model often prefers contamination-driven hazard categories when deception cues are implicit.

Product-category behavior. Product-category remains substantially harder than hazard-category. The model continues to over-predict broad, frequent categories such as *meat, egg and dairy products* and *prepared dishes and snacks*, producing high recall for these classes but low precision for many smaller categories. Several underrepresented classes still receive near-zero F1, which is consistent with the dataset's long-tail product distribution.

Takeaway. Grounded prompting with canonicalization is mainly a **reliability improvement** step: it reduces spurious failures caused by surface-form variations and provides a more faithful measurement of the underlying classification ability of the LLM. However, it does not address the fundamental long-tail generalization challenge, which motivates few-shot prompting (next section) and task-adaptive methods (fine-tuning and RAG).

3.1.6 Few-Shot Prompting (Targeted 4-shot)

Motivation and design choice (Why 4-shot?). After establishing a strong zero-shot baseline, we evaluated few-shot prompting to test whether a small amount of in-context supervision can reduce ambiguity and improve label selection—especially for the *product-category* task, which is substantially harder due to the larger label space (22 classes) and long-tail distribution. We selected a **4-shot** design for two practical reasons:

- **Hardware and runtime constraints.** In our local setup (Ollama with `llama3.1:8b`), increasing the number of examples directly increases prompt length, memory pressure, and end-to-end latency. This becomes a bottleneck for running hundreds of reports reproducibly on commodity hardware.
- **Targeted coverage over quantity.** Rather than adding many random examples, we used a small number of *targeted* examples to cover known weak spots observed in error patterns (rare hazard categories and frequently confused product categories). This yields better signal-to-context ratio and reduces the chance that irrelevant demonstrations bias the model.

Method. We build a fixed few-shot block from the training set (seeded for reproducibility) and prepend it to the same strict two-line output prompt. The 4 demonstrations are selected to address known failure modes:

1. A **fraud** example (hazard-category),
2. An **organoleptic aspects** example (hazard-category),
3. A **soups, broths, sauces and condiments** example (product-category),
4. A **cocoa and cocoa preparations, coffee and tea** example (product-category).

We retain the **strict parser with canonicalization** to ensure that minor output variants are not counted as errors. Prompts are passed via `stdin` (Windows-safe) with a timeout to avoid hangs on long contexts.

3.1.7 Results (Few-shot 4-shot)

Validation Set.

- Parse OK rate: 0.984
- Avg latency: 0.68 seconds/sample

Hazard-category:

- Accuracy: 0.878
- Macro-F1: 0.511
- Weighted-F1: 0.849

Product-category:

- Accuracy: 0.565
- Macro-F1: 0.449
- Weighted-F1: 0.576

Test Set.

- Parse OK rate: 0.978
- Avg latency: 0.71 seconds/sample

Hazard-category:

- Accuracy: 0.885
- Macro-F1: 0.555
- Weighted-F1: 0.866

Product-category:

- Accuracy: 0.601
- Macro-F1: 0.555
- Weighted-F1: 0.612

3.1.8 Analysis

Key effect: large gains on product-category. Compared to zero-shot (strict/canon), the targeted 4-shot setup yields a substantial improvement on *product-category*:

- Validation accuracy increases from ≈ 0.39–0.41 to 0.565,
- Test accuracy increases from ≈ 0.38–0.40 to 0.601,
- Macro-F1 also improves notably (validation: $\approx 0.38 \rightarrow 0.45$; test: $\approx 0.37 \rightarrow 0.55$).

This suggests that **in-context demonstrations primarily help the model learn the mapping from report phrasing to the correct product category**, reducing reliance on broad default categories and improving generalization to mid-frequency classes (e.g., *ices and desserts*, *seafood*, *soups, broths, sauces and condiments*).

Hazard-category: limited gains and mixed behavior. In contrast, hazard-category metrics do not improve consistently and can even decrease in macro-F1 on validation. This is expected because hazard-category is already relatively easy in zero-shot for dominant classes (*allergens*, *biological*, *foreign bodies*) and macro-F1 is heavily influenced by rare categories. In the few-shot results, *fraud* recall remains low and *migration* stays near-zero due to extreme scarcity (e.g., only 1 instance in the test set), meaning a small few-shot budget cannot reliably cover all long-tail hazards.

Reliability trade-off: prompt length vs. parse stability. Few-shot prompting increases context length and therefore slightly reduces parse success (Parse OK drops from ≈ 0.99 to ≈ 0.98) and increases latency. This reflects an inherent trade-off: demonstrations improve semantic alignment (especially for products) but require longer prompts, which can introduce formatting deviations and higher computational cost.

Takeaway. Targeted few-shot prompting is a strong "low-code" improvement path under compute constraints: it substantially improves *product-category* performance while keeping inference fully prompt-based (no parameter updates). However, addressing rare hazard categories robustly likely requires either more comprehensive in-context coverage (more shots) or model adaptation (supervised fine-tuning) and/or external knowledge injection (RAG), which we evaluate in subsequent sections.

3.2 Supervised Fine-Tuning (SFT)

Prompting-based baselines showed strong performance on the coarse hazard taxonomy but remained weak for product-category prediction, suggesting that the model can often detect the presence of a hazard (e.g., pathogens, allergens, foreign bodies) from explicit keywords, yet struggles to map diverse product descriptions into a 22-class taxonomy. To address this limitation, we next investigate supervised fine-tuning (SFT) using a transformer encoder and a classification head.

3.2.1 Multi-task fine-tuning setup

We first trained a multi-task model with a shared encoder and two task-specific classification heads, one for `hazard-category` (10 classes) and one for `product-category` (22 classes). We used `microsoft/deberta-v3-base` as the shared encoder and concatenated the `title` and `text` fields using a simple separator token. Inputs were truncated to a maximum length of 256 tokens, trained with batch size 16 for 4 epochs using AdamW (learning rate 2×10^{-5}, weight decay 0.01). To mitigate class imbalance, we used inverse-frequency class weights for cross-entropy loss in both tasks and applied linear warmup (6%) with a linear decay schedule. The overall training objective was the unweighted sum of the two losses:

$$\mathcal{L} = \mathcal{L}_{haz} + \mathcal{L}_{prod}. \tag{1}$$

The best checkpoint was selected by the average of the two macro-F1 scores on the validation set.

3.2.2 Multi-task results and diagnosis

The multi-task model improved substantially over early epochs for `hazard-category`, reaching a strong test accuracy of 0.900 and macro-F1 of 0.641. However, `product-category` performance remained poor, with test accuracy 0.336 and macro-F1 0.220.

This gap indicates that the shared representation learned by the encoder was dominated by the hazard objective. This behavior is consistent with the dataset characteristics: hazard prediction is often driven by explicit lexical cues (e.g., "Listeria", "Salmonella", "undeclared milk", "plastic"), while product prediction requires finer semantic normalization across many surface forms (e.g., "ready-to-eat meal" vs. "prepared dish", or multi-ingredient items). In a shared-encoder setting with equal loss weighting, gradient updates from the easier and more separable hazard task can dominate training, yielding a representation that is not sufficiently specialized for the more diverse product taxonomy. Moreover, the product label space is larger (22 vs. 10) and more long-tailed, which further increases its sample complexity and sensitivity to optimization dynamics.

3.2.3 Product-category single-task fine-tuning

To explicitly address the under-optimized `product-category` head, we trained a separate single-task classifier dedicated only to product prediction. The motivation was to (i) eliminate competition between tasks in the shared encoder, (ii) allow the model capacity and optimization trajectory to be fully shaped by product discrimination, and (iii) better exploit longer textual context when available. Under this single-task setting, product performance improved dramatically from macro-F1 0.220 (multi-task test) to macro-F1 0.534 (single-task test), with accuracy increasing from 0.336 to 0.717.

On the validation set, the best product model reached an accuracy of 0.719 and macro-F1 of 0.599, confirming that the product task benefits from dedicated optimization. These gains suggest that product prediction requires specialized semantic features that were not learned reliably under the multi-task objective. In practice, this also aligns with deployment considerations: if the downstream system prioritizes accurate product categorization (e.g., for product-specific risk reporting), a dedicated classifier can provide substantially better utility than a single shared multi-task model.

3.2.4 Summary of key findings

Overall, supervised fine-tuning yielded two distinct outcomes: (1) a strong hazard classifier can be obtained even under multi-task training (test macro-F1 0.641), and (2) product classification is significantly improved by moving to a single-task formulation (test macro-F1 0.534). This behavior highlights a practical trade-off between model simplicity (single shared encoder) and task-specific optimization, motivating our later discussion on when prompting-based baselines remain competitive versus when task-specific fine-tuning is necessary.

3.3 Retrieval-Augmented Generation (RAG)

While zero-shot prompting provides general reasoning capabilities and supervised fine-tuning (Section 3.2) improves task-specific discrimination, both approaches rely solely on parametric knowledge. To inject structured domain knowledge at inference time, we next investigate a Retrieval-Augmented Generation (RAG) strategy tailored to the recall-report classification setting.

3.3.1 Knowledge Base Construction

We constructed a lightweight knowledge base (KB) derived exclusively from the training split. The KB contains two types of documents for both tasks:

- **Definition documents**: One document per label, containing a short semantic definition of the category.
- **Example documents**: Up to 8 randomly sampled training examples per label (title + truncated text).

Hazard definitions were manually curated to reflect regulatory semantics (e.g., microbiological vs. chemical vs. foreign body). For product categories, we further enriched definitions by extracting frequent keywords and representative recall titles from up to 30 training samples per category. This produced a structured, label-centric knowledge base combining symbolic descriptions and empirical evidence.

3.3.2 Embedding and Retrieval Setup

All KB documents were embedded using `all-MiniLM-L6-v2` from Sentence-Transformers. Cosine similarity search was implemented via FAISS using normalized inner-product indexing.

Importantly, hazard and product documents were indexed separately, producing two retrieval indices. For each recall report, we retrieved top-k hazard documents and top-k product documents independently. To ensure semantic grounding, we enforced inclusion of at least one definition document among retrieved contexts whenever available.

The retrieved contexts were inserted into a constrained prompt template that:

- Explicitly enumerated allowed labels,
- Injected retrieved hazard knowledge,
- Injected retrieved product knowledge,
- Required strict one-line JSON output.

3.3.3 Validation Results and Ablation on k

We evaluated different retrieval depths ($k = 1, 3, 5$) on the validation set.

Validation (k=3) Hazard macro-F1: 0.306
Product macro-F1: 0.374
Parse success rate: 0.996

Validation (k=5) Hazard macro-F1: 0.300
Product macro-F1: 0.392

Validation (k=1) Hazard macro-F1: 0.302
Product macro-F1: 0.339

Increasing k slightly improved product performance but did not significantly affect hazard classification. This suggests that product categorization benefits from broader contextual grounding, while hazard prediction remains primarily driven by explicit lexical cues in the recall text.

3.3.4 Test Results

Using $k = 5$ (best validation performance for product), we obtained:

- **Hazard-category:** Accuracy 0.498, Macro-F1 0.261
- **Product-category:** Accuracy 0.458, Macro-F1 0.413
- **Parse success rate:** 0.986

Compared to zero-shot prompting, RAG improved product macro-F1 substantially (from $\sim$0.39 to 0.41 on test), but hazard macro-F1 remained modest. Compared to supervised fine-tuning, RAG remains inferior for hazard detection (SFT macro-F1 0.641) and also below single-task SFT for product (macro-F1 0.534).

3.3.5 Analysis

The observed behavior reflects structural differences between the two tasks:

- **Hazard-category** is often explicitly stated in recall descriptions (e.g., "Listeria", "Salmonella", "undeclared milk"), reducing reliance on external knowledge retrieval.
- **Product-category** requires semantic normalization across heterogeneous surface forms (e.g., "ready-to-eat meals", "prepared dishes", "frozen entrees"), where retrieval of label definitions and examples provides useful contextual anchors.

However, because the underlying LLM parameters remain frozen, retrieval alone cannot match the representational adaptation achieved by supervised fine-tuning. Thus, RAG provides moderate gains over pure prompting but does not fully close the gap to SFT.

3.3.6 Hybrid RAG: Task-Specific Retrieval Design

The vanilla RAG setup (Section 3.3) retrieves both hazard-related and product-related contexts from a shared knowledge base containing (i) label definitions and (ii) training examples. Although this can improve semantic grounding for product categorization, it may also introduce noise, especially for hazard classification where explicit lexical cues (e.g., "Listeria", "Salmonella", "undeclared milk") are often sufficient.

Therefore, we investigated a **hybrid RAG** design motivated by the asymmetric nature of the two subtasks:

- **Hazard-category retrieval is constrained to definitions only.** The goal is to avoid misleading example-level overlaps across hazard classes and to provide compact, symbolic grounding rather than noisy evidence.
- **Product-category retrieval uses both definitions and examples.** This is because product labels often require semantic normalization (e.g., mapping diverse surface forms such as "ready-to-eat", "meal", "sauce" to the correct category), where example-based retrieval is expected to help.

Hybrid KB and Retrieval Implementation We constructed KB documents from the training split, including hazard definitions, product definitions, and up to 8 examples per label. Two FAISS indices were built using Sentence-Transformer embeddings (`all-MiniLM-L6-v2`):

- A **hazard index** containing only hazard definition documents.
- A **product index** containing product definition + example documents.

At inference time, we retrieved $k_h = 1$ hazard-definition document and $k_p = 5$ product documents, injected them into the prompt, and constrained the model to output a single-line JSON with labels restricted to the predefined label space.

Results On the validation set, the hybrid configuration achieved:

- **Hazard-category:** Macro-F1 = 0.185 (Accuracy = 0.368)
- **Product-category:** Macro-F1 = 0.379 (Accuracy = 0.412)
- **Parse success rate:** 0.996

On the test set ($k_h = 1, k_p = 5$):

- **Hazard-category:** Macro-F1 = 0.178 (Accuracy = 0.362)
- **Product-category:** Macro-F1 = 0.382 (Accuracy = 0.448)
- **Parse success rate:** 0.990

Discussion of Hybrid Behavior Contrary to the initial motivation, restricting hazard retrieval to definitions *degraded* hazard classification substantially compared to vanilla RAG (test Macro-F1: 0.178 vs. 0.261). This suggests that, for hazard prediction, example-level retrieval (even if noisy) provides useful lexical anchors that improve alignment between the report phrasing and the label space. In other words, hazard classification benefits more from **pattern matching to retrieved recall examples** than from abstract definitions alone.

For product categorization, performance remained comparable to vanilla RAG (test Macro-F1 $\approx$ 0.38–0.41), indicating that product prediction is relatively robust to the hybrid change and continues to benefit modestly from retrieval-based semantic grounding.

Overall, hybrid RAG confirms that retrieval design must be **task-dependent**, but the direction is not always intuitive: for this dataset, hazard retrieval appears to require concrete examples rather than solely symbolic label definitions.

4 Evaluation and Comparative Analysis

All approaches—prompt-based inference, supervised fine-tuning (SFT), and Retrieval-Augmented Generation (RAG)—were evaluated under a consistent experimental setup using the official train/validation/test splits of the SemEval-2025 Task 9 dataset.

We report Accuracy, Macro-F1, Micro-F1, and Weighted-F1. Given the strong class imbalance and long-tail distribution of both hazard and product categories, **Macro-F1** is treated as the primary evaluation metric, as it better reflects performance on minority classes.

4.1 4.1 Quantitative Comparison Across Paradigms

Table 5 summarizes the test-set performance of all major configurations.

Table 5: *Test-set performance comparison across paradigms*

Method	Hazard Macro-F1	Product Macro-F1
Zero-shot Prompting	∼0.26–0.57 (varies by prompt)	∼0.39–0.48
RAG (k=5)	0.261	0.413
Hybrid RAG	0.178	0.382
SFT (Multi-task)	0.641	0.220
SFT (Product-only)	–	0.534

Several important patterns emerge:

- **SFT dominates hazard classification**, achieving a test macro-F1 of 0.641—more than double the RAG-based approaches.
- **Product classification benefits most from dedicated SFT**, where a single-task product model reaches 0.534 macro-F1.
- RAG improves product performance compared to pure prompting, but remains substantially below supervised fine-tuning.
- Hybrid RAG, despite its task-specific retrieval design, degrades hazard performance.

4.2 Hazard vs. Product: Asymmetric Task Behavior

The two subtasks exhibit fundamentally different behavior:

Hazard-category Hazard detection often relies on explicit lexical cues in recall reports (e.g., "Listeria", "Salmonella", "undeclared milk", "plastic fragments"). Because these cues are directly observable in the text, parametric adaptation through SFT enables the model to form strong lexical-to-label mappings.

Retrieval augmentation offers limited benefit, as hazard definitions are abstract and example retrieval does not significantly enrich the signal beyond what is already present in the report text.

Product-category Product categorization requires semantic normalization across heterogeneous surface forms (e.g., "ready-to-eat meal", "prepared dish", "chicken entrée"). Here, retrieval provides moderate gains by anchoring the report to similar training examples. However, full adaptation via supervised fine-tuning is necessary to learn robust semantic boundaries between 22 product categories.

4.3 Rare-Class and Long-Tail Behavior

Macro-F1 differences highlight the long-tail challenge:

- Prompting and RAG struggle with minority hazard classes such as `migration`, `organoleptic aspects`, and `packaging defect`.
- Multi-task SFT substantially improves minority hazard categories, including better recall for `packaging defect` and `other hazard`.
- Product-only SFT significantly improves minority product classes compared to multi-task training, indicating that gradient competition in the shared encoder suppressed product learning.

These findings demonstrate that **parameter adaptation is critical for rare-class robustness** in safety-critical NLP tasks.

4.4 Structured Error Analysis

We identify four dominant failure modes:

1. **Hazard Confusion:** Misclassification between biologically and chemically related hazards, particularly when recall descriptions are short or ambiguous.
2. **Product Boundary Ambiguity:** Confusion between semantically close categories (e.g., "prepared dishes and snacks" vs. "meat, egg and dairy products").
3. **Keyword Over-reliance:** Prompt-based systems often anchor on a single token (e.g., "milk") without fully interpreting context.

4. **Definition Over-generalization (Hybrid RAG):** Using hazard definitions alone removes example-level lexical anchors, reducing alignment between report phrasing and label semantics.

Importantly, high parse-success rates in RAG-based systems ($\sim$0.99) do not necessarily imply high classification quality. Format compliance and decision accuracy are distinct properties.

4.5 Computational Trade-offs

Each paradigm presents distinct trade-offs:

- **Prompting:** Zero training cost, moderate inference latency, limited rare-class robustness.
- **RAG:** No training cost but higher inference latency due to embedding + retrieval; modest gains for product categories.
- **SFT:** Higher computational cost during training, but superior classification accuracy and stable inference.

For safety-critical applications such as food hazard detection, where minority-class recall is essential, supervised fine-tuning provides the most reliable performance, despite its higher training cost.

5 Discussion

This study explored three major paradigms for food hazard detection using LLMs: prompt-based inference, supervised fine-tuning (SFT), and Retrieval-Augmented Generation (RAG). The results reveal important insights about reliability, robustness, and practical deployment trade-offs in safety-critical NLP systems.

5.1 When Is Prompting Enough?

Zero-shot and few-shot prompting demonstrated surprisingly strong performance for majority hazard classes such as `allergens` and `biological`. This is largely due to the presence of explicit lexical triggers (e.g., "Listeria", "Salmonella", "undeclared milk") in recall reports.

However, prompting consistently struggled with:

- Rare hazard categories (e.g., `migration`, `organoleptic aspects`)
- Ambiguous product boundaries
- Subtle regulatory violations (e.g., `fraud`)

Although prompting requires no training cost and maintains interpretability, its instability across minority classes limits its suitability for high-stakes regulatory applications.

5.2 Why Supervised Fine-Tuning Outperformed Retrieval

Supervised fine-tuning yielded the strongest hazard classification performance (Macro-F1 = 0.641). This confirms that parameter adaptation enables the model to internalize domain-specific mappings beyond surface-level keyword matching.

Notably, multi-task training improved hazard prediction but degraded product classification. When trained separately, product-only SFT significantly improved Macro-F1 (0.534), indicating gradient competition between tasks in the shared encoder.

These findings suggest that:

- Hazard classification benefits from strong lexical alignment.
- Product classification requires deeper semantic boundary learning.
- Multi-task setups may require architectural adjustments (e.g., task-specific heads or balancing strategies).

5.3 Why RAG Did Not Improve Hazard Performance

Contrary to expectations, RAG did not improve hazard detection and in some cases reduced performance.

Two key factors explain this:

1. **Redundancy of Retrieved Knowledge:** Hazard definitions are often abstract and do not add discriminative power beyond explicit textual cues already present in recall reports.
2. **Retrieval Noise:** Increasing k introduces semantically related but label-misaligned examples, which can bias the generation process.

Hybrid RAG, which restricted hazard retrieval to definitions only, further reduced hazard Macro-F1. This suggests that removing example-level lexical grounding weakens label alignment rather than strengthening conceptual reasoning.

For product classification, RAG provided modest improvements over pure prompting, indicating that semantic anchoring to similar examples can be helpful when surface forms vary.

5.4 Safety-Critical Implications

Food hazard detection is inherently safety-critical. In such settings:

- High Macro-F1 is more important than overall accuracy.
- Rare-class recall is operationally significant.
- Stable decision boundaries are preferable to flexible generative reasoning.

From this perspective, supervised fine-tuning offers the most reliable foundation for deployment.

Prompting and RAG may serve as rapid prototyping or low-resource solutions, but they exhibit higher variance across minority classes.

5.5 Practical Constraints and Engineering Trade-offs

Several implementation decisions were influenced by computational constraints:

- The experiments used lightweight embedding models (all-MiniLM-L6-v2) to enable efficient retrieval under limited GPU memory.
- Instruction-tuned LLM inference was performed locally, requiring careful control of generation length and temperature.
- Retrieval depth (k) was limited to avoid latency explosion and memory overhead.

Under greater computational resources, future work could explore:

- Larger embedding models for higher retrieval precision
- Cross-encoder reranking for improved passage selection
- Joint retriever-generator fine-tuning
- Multi-task SFT with improved task balancing strategies

Thus, some design decisions represent engineering trade-offs rather than theoretical limitations of the paradigms.

5.6 Key Takeaways

The main insights from this study are:

1. Prompting captures majority-class lexical hazards effectively but lacks long-tail robustness.
2. Retrieval does not automatically improve reliability; its benefit depends on knowledge quality and task structure.
3. Parameter adaptation via supervised fine-tuning remains the most effective strategy for safety-critical classification.
4. Multi-task learning introduces task interference that must be carefully managed.
5. High parse-compliance does not imply high classification quality.

Overall, the results demonstrate that while LLM-based prompting and RAG offer flexible and low-training-cost solutions, supervised fine-tuning provides superior stability and rare-class performance in structured hazard detection tasks.

6 Conclusion

This project investigated the application of Large Language Models (LLMs) to food hazard detection under three paradigms: prompt-based inference, supervised fine-tuning (SFT), and Retrieval-Augmented Generation (RAG). Through systematic experimentation on the SemEval-2025 Task 9 dataset, we analyzed performance differences, rare-class robustness, and computational trade-offs in a safety-critical classification setting.

The results show that prompt-based inference can capture majority hazard classes effectively when explicit lexical cues are present. However, its performance degrades significantly on minority categories and semantically ambiguous product classes. While RAG introduces external knowledge and structured context, its impact depends heavily on knowledge quality and retrieval alignment. In this task, retrieval provided only modest gains for product classification and failed to improve hazard detection, highlighting that external knowledge is not universally beneficial in structured regulatory domains.

Supervised fine-tuning consistently delivered the most robust hazard classification performance, particularly for long-tail and rare categories. Product-only SFT further demonstrated that task-specific adaptation is crucial when semantic category boundaries are subtle. These findings reinforce the importance of parameter adaptation in safety-critical NLP systems where reliability across all classes—not only majority ones—is essential.

Importantly, this study reveals that high format compliance (parse success) does not guarantee classification quality, and that architectural choices such as multi-task training can introduce task interference effects. Moreover, computational constraints influenced some engineering decisions, including the use of lightweight embedding models and limited retrieval depth. Future work could explore larger retrievers, reranking strategies, and better task-balancing mechanisms to further enhance performance.

Overall, this comparative analysis demonstrates that while prompting and RAG provide flexible and low-cost deployment options, supervised fine-tuning remains the most reliable approach for structured food hazard detection. In safety-critical regulatory environments, stability and rare-class robustness outweigh the convenience of parameter-free solutions.

This work highlights the broader lesson that LLM paradigms must be selected not only based on novelty or convenience, but on task structure, class distribution, and operational reliability requirements.

A Detailed Experimental Results

Table 6: *Comprehensive test-set performance comparison*

Method	Parse Rate	Haz Macro-F1	Prod Macro-F1	Avg Latency (s)
Zero-shot Strict	0.991	0.598	0.388	0.676
RAG (k=5)	0.986	0.261	0.413	∼3.2
Hybrid RAG	0.990	0.178	0.382	∼3.0
SFT (Multi-task)	1.000	0.641	0.220	∼0.01
SFT (Product-only)	1.000	–	0.534	∼0.01

Table 7: *Effect of retrieval depth k on validation performance*

Configuration	Haz Macro-F1	Prod Macro-F1	Parse Rate
RAG (k=1)	0.302	0.339	0.996
RAG (k=3)	0.306	0.374	0.996
RAG (k=5)	0.300	0.392	0.996
Hybrid (k_h=1, k_p=5)	0.185	0.379	0.996

Table 8: *Observed behavior on minority hazard categories (qualitative summary)*

Hazard Category	Prompting	RAG	SFT
migration	Very Low	Very Low	Improved
organoleptic aspects	Low	Low	Moderate
packaging defect	Low	Low	Strong Improvement
other hazard	Low	Low	Improved

B Reproducibility and Configuration Details

All experiments were conducted using the official SemEval-2025 Task 9 dataset splits.

Prompt-based and RAG experiments used the instruction-tuned LLM `llama3.1:8b` with temperature set to 0 for deterministic output.

RAG retrieval used the `all-MiniLM-L6-v2` embedding model with cosine similarity implemented via FAISS inner-product search over normalized vectors.

Supervised fine-tuning was implemented using a pretrained transformer backbone with standard cross-entropy loss and early stopping based on validation Macro-F1.

Random seeds were fixed where applicable to ensure reproducibility.