OPI-DRO-HEL at SemEval-2025 Task 9: Integrating Transformer-Based Classification with LLM-Assisted Few-Shot Learning for Food Hazard Detection

Martyna Śpiewak, Daniel Karaś

National Information Processing Institute
{mspiewak,dkaras@opi.org.pl}

Abstract

In this paper, we propose a hybrid approach for food hazard detection that combines a fine-tuned RoBERTa classifier with few-shot learning using an LLM model (GPT-3.5-turbo). We address challenges related to unstructured text and class imbalance by applying class weighting and keyword extraction (KeyBERT, YAKE, and Sentence-BERT). When RoBERTa's confidence falls below a given threshold, a structured prompt which comprising the title, extracted keywords, and a few representative examples is used to re-evaluate the prediction with ChatGPT.

1 Introduction

The SemEval-2025 Food Hazard Detection task (Randl et al., 2025) focuses on automatically identifying mentions of food hazards, such as allergens or chemical contaminants, in textual data. This challenge is critical for public health, as rapid and accurate detection can help prevent outbreaks and protect consumers.

Our system employs a two-stage classification pipeline. First, we use a RoBERTa-based classifier (Liu et al., 2019), fine-tuned on the training data, to predict whether a document includes a food hazard. If the model's confidence is below a given threshold, we move on to a secondary check with an LLM model. In this step, we provide ChatGPT with the document's title, a list of keywords, and a few examples of how to identify hazards — a few-shot learning setup (Snell et al., 2017; Wang et al., 2020; Brown et al., 2020; Schick and Schütze, 2021). This additional step helps detect subtle or rare hazard mentions that RoBERTa might overlook.

Despite these efforts, our system ranked 17th out of 27 teams. Further analysis showed that our approach struggled with complex language describing hazards. In addition, the automatic extraction of keywords and the variability in ChatGPT's responses sometimes led to different outcomes. These results suggest that improved keyword selection, more precise threshold tuning, and better domain guidance for ChatGPT could lead to higher performance. Our complete codebase, including preprocessing and training scripts, is publicly available at: https://gitlab.com/mspiewak/food-hazard-detection.

2 Background

The SemEval-2025 Food Hazard Detection task provides a structured dataset for identifying food hazard information in text documents. Each record in the dataset contains metadata (e.g., year, month, day, country), a title, and a full-length text field that describes a food recall event. The labels include both product and hazard each of which belongs to a higher-level category: product-category (22 possible categories) and hazard-category (10 possible categories).

2.1 Related work

Natural Language Processing (NLP) has played a major role in automating food hazard detection using text classification and information extraction methods. Recent studies have used large language models and deep learning techniques to improve food safety monitoring.

One key study, CICLe (Randl et al., 2024), presented a dataset of food recall announcements and compared NLP models like RoBERTa and XLM-R with traditional machine learning techniques. They found that logistic regression using tf-idf features outperformed transformer models in low-resource settings, highlighting the need for efficient, flexible methods. Motivated by these findings, our approach uses conformal prediction techniques to improve classification while reducing computational costs.

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 1174–1180

Other related studies, such as (Özen et al., 2024), have applied LLMs to extract chemical hazards from scientific literature with high accuracy. Similarly, (Nogales et al., 2022) demonstrated the effectiveness of deep learning with categorical embeddings for predicting food safety risks from European Union data.

3 System overview

The SemEval-Task combines two sub-tasks: (ST1) text classification for food hazard prediction, predicting the type of hazard and product, and (ST2) food hazard and product "vector" detection, predicting the exact hazard and product.

This task prioritizes accurate hazard detection and employs a two-step evaluation metric based on the macro F1 score, emphasizing hazard classification in both sub-tasks. In this study, we focused exclusively on ST1, developing a classification system tailored to the prediction of food hazards and associated product categories.

3.1 Data Analysis

The dataset is divided into three subsets: training, validation, and test sets. The training set consists of 5,082 records and is used for model development, learning and fine-tuning hyperparameters. The validation set, comprising 565 records, and the test set, containing 997 records, serves as an independent evaluation dataset to assess the generalization performance of the model.

Hazard Category (10 classes)	Train (%)	Notes
allergens	36.48%	comparable across splits
biological hazards	34.26%	comparable across splits
foreign bodies	11.04%	comparable across splits
fraud	7.30%	comparable across splits
chemical hazards	5.65%	minor variations observed
microbiological hazards	2.00%	comparable across splits
physical contaminants	1.00%	comparable across splits
other hazards	1.00%	comparable across splits
food additives & flavorings	0.47%	underrepresented in all splits
migration	0.06%	underrepresented in all splits

Table 1: Distribution of hazard categories (10 classes) in the training set, sorted by percentage (descending).

A strong imbalance is observed in the distribution of both the hazard category and product category variables. Tables 1 and 2, highligh the difficulty of detecting underrepresented categories. The hazard category is dominated by allergens and biological hazards, which together account for over 70% of all cases. Other categories, such as foreign bodies, fraud, and chemical hazards, appear less frequently, while some categories like migration and food additives and flavorings are particularly underrepresented.

Similarly, the product category variable shows a high concentration in a few categories. Meat, egg, and dairy products represent the most frequently reported category, followed by cereals and bakery products and fruits and vegetables. Other categories, such as sugars and syrups, feed materials, food contact materials, and honey and royal jelly, are significantly less common.

Product Category (22 classes)	Train (%)	Notes
meat, egg & dairy products	28.22%	minor variations observed
cereals & bakery products	13.20%	comparable across splits
fruits & vegetables	10.53%	comparable across splits
prepared dishes & snacks	9.23%	comparable across splits
seafood	5.27%	minor variations observed
soups, broths, sauces & condiments	5.19%	minor variations observed
nuts, nut products & seeds	5.16%	comparable across splits
ices & desserts	4.37%	comparable across splits
cocoa & cocoa preparations, coffee & tea	4.13%	minor variations observed
confectionery	3.35%	minor variations observed
non-alcoholic beverages	2.64%	minor variations observed
dietetic foods, food supplements, fortified foods	2.58%	comparable across splits
herbs & spices	2.46%	minor variations observed
alcoholic beverages	1.16%	comparable across splits
other food product / mixed	1.08%	comparable across splits
pet feed	0.39%	minor variations observed
fats & oils	0.37%	underrepresented in all splits
food additives & flavourings	0.16%	underrepresented in all splits
honey & royal jelly	0.16%	underrepresented in all splits
food contact materials	0.14%	underrepresented in all splits
feed materials	0.12%	underrepresented in all splits
sugars & syrups	0.10%	underrepresented in all splits

Table 2: Distribution of product categories (22 classes) in the training set, sorted by percentage (descending).

The distribution of categories is generally consistent across dataset splits; however, minor discrepancies, such as differences in the proportions of cocoa, coffee, and confectionery, could introduce subtle biases. The underrepresentation of rare categories may limit the model's ability to generalize to less common incidents.

The analysis will concentrate on the title and text fields, which are unstructured and irregular. Because these fields lack a consistent format, it is not possible to automatically extract discrete features from them. Representative examples of frequently occurring textual patterns are provided in Table 3. Instead, we examine the complete content using text analysis techniques. This approach provides a comprehensive understanding of the underlying content and the linguistic patterns present in the data.

FOR IMMEDIATE RELEASE – CINCINNATI, Ohio, April 7, 2008 – Inter-American Products, Inc., a division of The Kroger Co., ... Sell By date of December 3, 2008. The two codes are: DEC0308 8070 and DEC0308 8080...
Food Recall Warning (Allergen) – Coconut Town brand Coconut Cream Powder recalled due to undeclared milk. **Recall date**: November 29, 2016. **Reason for recall**: Allergen – Milk...
PRA No. 1998/3436. Date published 14 Jan 1998. **Product description/Brands**: Gibbs 400g, Sumners 350g & Foodlands 350g...
Updated Food Recall Warning – Coconut Tree brand Shredded Young Coconut recalled due to Salmonella. **Recall date**: January 28, 2018. **Reason for recall**: Microbiological – Salmonella...

Table 3: Sample text entries from the training dataset.

3.1.1 Text Preprocessing Strategy

We applied a focused preprocessing to standardize the text fields prior to model training. Key steps included:

- **HTML Removal** Stripping out HTML tags to eliminate irrelevant markup.
- **Whitespace and Special Character Normalization** eplacing non-breaking spaces with regular spaces and consolidating multiple spaces and line breaks.
- **Case Conversion** Converting text to lowercase to ensure consistency.
- **Numeric and URL Removal** Eliminating numbers and hyperlinks that rarely convey hazard information
- **Stopword Elimination (Optional)** Removing common English stopwords based on experimental settings.

This cleaning pipeline reduces data variability and ensures that the model focuses on meaningful terms related to food hazards and products.

3.2 Two-Step Classification Pipeline

Our approach combined a fine-tuned RoBERTa model with an LLM model in a sequential classification setup. First, we trained the RoBERTa classifier on the cleaned text field of each record. After obtaining a probability estimate for the predicted class, we compared it to a threshold. If RoBERTa's confidence exceeded this threshold, we accepted its prediction as final.

However, whenever the probability was below the threshold, we triggered a secondary check. In this step, we extracted keywords from the text and the recall notice's title are used to construct a prompt. We then used few-shot learning promps with ChatGPT, providing a small set of labeled examples to guide the model in predicting the hazard category. This two-stage process harnesses RoBERTa's high precision for clear-cut cases while leveraging ChatGPT's contextual understanding for ambiguous instances.

4 Experimental setup

To ensure that our findings can be reproduced, we detail our experimental design below. The dataset is divided into 5,082 training records for model development and hyperparameter tuning, 565 validation records for model selection, and 997 test records for independent evaluation. Our text preprocessing pipeline involves removing HTML tags, normalizing whitespace, converting all text to lowercase, and eliminating numeric elements and URLs, with an optional step for stopword removal as needed.

4.1 RoBERTa-Based Classification with Imbalance Handling

The training dataset is highly imbalanced, with some hazard categories appearing much more frequently than others. This imbalance can cause the model to favor majority classes and perform poorly on rare hazards. To fix this, we added class weighting to the cross-entropy loss function, giving more emphasis to underrepresented categories.

Model Architecture and Training Setup We fine-tuned a RoBERTa model for both hazard and product category classification, treating each as a multi-class classification task. Importantly, the same architecture is employed for both variables. For hazard classification, the model's output layer was configured with 10 neurons corresponding to the 10 hazard categories, and similarly, the product category classification used an output layer sized to the number of product categories. To handle imbalance in the hazard data, we computed class weights using the formula:

$$w_c = \frac{N}{k \times N_c}$$

where N is the total number of samples, N_c is the number of samples in class c, and k is the total

number of unique classes. This "balanced" weighting scheme is inspired by the approach described in (King and Zeng, 2001), which proposes adjusting the loss function to counteract the bias introduced by imbalanced datasets.

Our structured training approach included:

- **Tokenization and Encoding** We used RoBERTa's subword tokenization (Sennrich et al., 2016) to break words into smaller units, which helped preserve technical terms related to food hazards and handle rare or domain-specific words effectively.
- **Optimization Strategy** The model was trained using AdamW with weight decay (Loshchilov and Hutter, 2019) to prevent overfitting. A linear learning rate scheduler was applied to gradually reduce the learning rate, stabilizing updates and improving convergence.
- **Hyperparameter Tuning** We optimized learning rate, batch size, dropout rate, weight decay, and warmup steps using Optuna (Akiba et al., 2019), selecting the best configuration based on macro F1-score.

Due to the severe imbalance in the data, accuracy can be misleading since it is often dominated by the majority classes. Instead, we used the macro F1-score to fairly evaluate performance on both common and rare categories. This metric helped the model better identify rare hazards and improved recall in critical cases.

4.2 Keyword Extraction Process

To reduce the input length for few-shot learning and focus the prompts, we applied three keyword extraction methods – KeyBERT, YAKE, and Sentence-BERT – after cleaning each text. We extracted the top ten keywords for every text using each method:

- **KeyBERT** (Grootendorst, 2020) uses transformer-based embeddings to identify the terms most relevant to a given document's content.
- **YAKE (Campos et al., 2020)** employs an unsupervised, statistical method, ranking words based on frequency and positional features.
- **Sentence-BERT** (Reimers and Gurevych, 2019) considers semantic similarities at the sentence level, pinpointing contextually important expressions.

These keywords summarize key concepts, ensuring that an LLM model prompts remain targeted and manageable.

4.3 Few shot learning

To improve the classification accuracy for low-confidence predictions, we implemented a few-shot learning approach using ChatGPT (GPT-3.5-turbo) along with the RoBERTa classifier. We set the threshold at 0.5, corresponding to the median confidence of correct validation predictions, which strikes a balance between avoiding unnecessary LLM calls and capturing genuinely uncertain cases. Samples below this threshold are then re-evaluated by the LLM. In this case, a structured prompt is generated that includes the recall notice's title, a list of extracted keywords, and a few representative examples from the training set (selected based on cosine similarity of tf-idf embeddings). For each extraction method (KeyBERT, YAKE, Sentence-BERT), we collect only the top ten keywords returned by that method. We do not concatenate all 30 keywords into a single pool, nor do we remove duplicates across methods, since each method's output is used in a separate prompt configuration. Within each prompt, the keywords appear in the exact order provided by the extractor – reflecting their ranking – rather than as an unordered bag-of-words. This preserves the method, the specific context and ordering that proved most effective in our experiments. ChatGPT then predicts the most appropriate hazard class from a predefined list of valid classes. If ChatGPT's prediction matches one of these classes, it replaces the original RoBERTa output; otherwise, the RoBERTa prediction is retained.

All few-shot prompts follow a consistent three-part structure. First, we include a brief task description that outlines the classification objective. Next, we present N labeled examples in the format: `Title + Keywords` $\rightarrow$ `Label`. Finally, the prompt ends with the target query: `Title + Keywords` $\rightarrow$ `?`. To assess the impact of prompt design, we tested minor variations – such as ordering the examples by cosine similarity – and found that prompt ordering could change the macro F1 score by up to one point. We plan to perform a more extensive prompt-engineering study in future work.

	RoBERTa	FSL & KeyBERT	FSL & YAKE	FSL & Sentence-BERT
Without Class Balancing				
Validation Dataset				
hazard-category	0.86654	0.86077	0.86308	0.86155
product-category	0.67434	0.68555	0.68603	0.6963
Test Dataset				
hazard-category	0.77584	0.77607	0.77537	0.77948
product-category	0.67553	0.74464	0.71664	0.75418
With Class Balancing				
Validation Dataset				
hazard-category	0.88952	0.89560	0.85883	0.88107
product-category	0.70174	0.71052	0.71002	0.71084
Test Dataset				
hazard-category	0.76091	0.76644	0.75896	0.76644
product-category	0.70422	0.77889	0.74011	0.77578

Table 4: Comparison of classification methods (F1 macro scores) for hazard and product categories on the validation and test datasets. Results are shown for models built without and with class balancing.

Our implementation utilizes Huggingface Transformers (v4.49.0) (Hugging Face Inc., n.d.; Wolf et al., 2020), Optuna (v4.2.1), OpenAI (v1.60.1) (OpenAI, 2022), KeyBERT (v0.9.0), Yake (0.4.8), Sentence Transformers (v3.4.1) and other standard Python libraries (numpy, pandas, scikit-learn).

5 Results

Table 4 presents the performance of our system with and without class balancing, measured in macro F1-scores on both the validation and test datasets. Our submission to the competition used class balancing, as adding class weights improved performance for several categories. With class balancing, the few shot learning (FSL) approach combined with KeyBERT achieved the highest score for hazard-category classification (0.89560), while all few-shot learning variants improved upon the baseline RoBERTa for product-category classification, with FSL using Sentence-BERT reaching 0.71084. On the test dataset, however, the performance for hazard-category classification is notably lower, with scores ranging from 0.75896 to 0.76644, compared to 0.88952–0.89560 on the validation set. For product-category classification, the best result on the test set was 0.77889, which also represents a modest improvement over the RoBERTa baseline.

We submitted the configuration using RoBERTa with KeyBERT. However, the gap between validation and test results for hazard-category classification suggests possible overfitting, while we do not observe this for product-category classification. Notably, for product-category classification on the test set, methods using the LLM component clearly improved performance, highlighting the benefit of few-shot learning.

Our experiments show that adding few-shot learning with targeted keyword extraction improves the basic RoBERTa performance. Using class balancing, especially for product categories, effectively addresses data imbalance. Additionally, KeyBERT and Sentence-BERT work better than YAKE, providing more reliable support for the few-shot component.

The main source of errors is that underrepresented classes tend to be misclassified with high confidence. In many instances, rare hazard and product categories are incorrectly predicted with strong probabilities, indicating that the model is overly biased toward majority classes. Additionally, the few-shot learning component using ChatGPT struggles with these cases because the examples provided in the prompts are insufficient or weak, leading to more errors for rare classes. These observations suggest that both the primary model and the LLM require improved strategies, such as improved prompt design and better handling of low-frequency classes, to effectively address these issues.

5.1 Limitations

While our approach shows promise, it did not perform consistently well and seems overfitted to the training set. Here, we discuss the main challenges and possible improvements. The imbalance in category distributions can hurt model generalization, but it could be improved by data augmentation and re-sampling methods like stratified sampling. Also, using the validation set for extra training could help the model adapt better, especially in low-data scenarios. Another area to improve is selecting can-

didates for few-shot learning, where approaches like active learning or uncertainty-based sampling might help. Fixing these issues through better data handling and selection methods would make the model more robust and improve its performance.

6 Conclusion

We have developed a two-stage classification system for food hazard detection that combines the strengths of both RoBERTa and ChatGPT to tackle challenges posed by imbalanced data. Our approach employs class weighting to reduce bias toward majority classes and leverages few-shot learning to improve predictions when the model's confidence is low. Although our system ranked 17th out of 27 teams, indicating room for improvement, error analysis indicates that underrepresented classes are still frequently misclassified with high confidence. In future work, we plan to improve the prompt design and use active learning strategies to improve model adaptability. Additionally, we will explore advanced data augmentation techniques to better capture the characteristics of rare hazard categories. These improvements should make the model more robust and generalizable, improving its effectiveness in real-world food safety applications.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. *Preprint*, arXiv:1907.10902.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

Hugging Face Inc. n.d. Hugging face. `https://huggingface.co`. Accessed: Month Day, Year.

Gary King and Langche Zeng. 2001. Logistic regression in rare events data. *Political Analysis*, 9:137 – 163.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Alberto Nogales, Rodrigo Díaz-Morón, and Álvaro J. García-Tejedor. 2022. A comparison of neural and non-neural machine learning models for food safety risk prediction with european union rasff data. *Food Control*, 134:108697.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. `https://openai.com/blog/chatgpt`. Accessed: Month Day, Year.

Korbinian Randl, John Pavlopoulos, Aron Henriksson, and Tony Lindgren. 2024. CICLe: Conformal in-context learning for largescale multi-class food risk classification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7695–7715, Bangkok, Thailand. Association for Computational Linguistics.

Korbinian Randl, John Pavlopoulos, Aron Henriksson, Tony Lindgren, and Juli Bakagianni. 2025. SemEval-2025 task 9: The food hazard detection challenge. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, Vienna, Austria. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. *Preprint*, arXiv:2009.07118.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. *Preprint*, arXiv:1703.05175.

Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Brew, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45. Association for Computational Linguistics.

Neris Özen, Wenjuan Mu, Esther D. van Asselt, and Leonieke M. van den Bulk. 2024. Extracting chemical food safety hazards from the scientific literature automatically using large language models. *Preprint*, arXiv:2405.15787.