madhans476 at SemEval-2025 Task 9: Multi-Model Ensemble and Prompt-Based Learning for Food Hazard Prediction

Madhan S[1], **Gnanesh A R**[1], **Gopal**[1] **and Sunil Saumya**[1]
[1]Department of Data Science and Artificial Intelligence,
Indian Institute of Information Technology Dharwad, Dharwad, Karnataka, India
(22bds036, 22bds023, 22bds025, sunil.saumya)@iiitdwd.ac.in

Abstract

Food safety is a critical public health concern requiring rapid and accurate identification of potential hazards in food products. This paper presents our approach to SemEval-2025 Task 9, the Food Hazard Detection Challenge, which focuses on automatically classifying and extracting hazard information from food recall notifications. We propose a hybrid system combining traditional machine learning with state-of-the-art language models, implementing an ensemble approach for hazard classification (Sub-Task 1) and a prompt-engineered extraction method using Flan-T5-XL for precise hazard and product detection (Sub-Task 2). The results demonstrate the effectiveness of combining multiple complementary models while highlighting challenges in exact vector matching for food safety applications.

1 Introduction

Food safety incidents can have severe consequences for public health and the food industry, making rapid and accurate identification of food hazards crucial. The SemEval-2025 Task 9: Food Hazard Detection Challenge (Randl et al., 2025) addresses this critical need by focusing on automated analysis of food recall notifications, aiming to classify and extract specific hazard information from text descriptions. This task builds upon the CICLe dataset (Randl et al., 2024), which provides a comprehensive collection of food recall notifications.

Our approach to this challenge combines the strengths of traditional machine learning(ML) techniques with Large Language Models(LLMs). For Sub-Task 1 (ST1), we implemented a novel ensemble system integrating XGBoost (Chen and Guestrin, 2016) with fine-tuned versions of GPT-2 Large (Radford et al., 2019) and LLaMA 3.1 1B (Touvron et al., 2023) models. The Sub-Task 2 (ST2) utilizes a prompt-engineered approach with Flan-T5-XL (Chung et al., 2022), focusing on precise extraction of hazard and product information. This hybrid approach allows us to leverage both the statistical power of traditional methods and the semantic understanding capabilities of LLMs.

We achieved competitive results in both conception and evaluation phases, with F1-scores of 0.78 and 0.74 respectively for ST1. In contrast, ST2 was more challenging with an F1-score of 0.05, reflecting the difficulty of exact vector matching in food safety. Our model performed well on common hazard categories but struggled with rare classes and precise match accuracy.The complete codebase for our system is available at https://github.com/madhans476/Food-hazard-detection-SEMEVAL-2025.git.

2 Background

The data set for this task is derived from the CICLe corpus (Randl et al., 2024), a large-scale dataset of 7,546 English food recall notices annotated with hazard types (e.g., biological, chemical, physical) and product categories (e.g., dairy, meat, beverages). Its broad coverage of food safety incidents makes it well-suited for training and evaluating NLP models for food hazard detection.

2.1 Related Work

Food safety monitoring using NLP has gained attention for its potential to automate hazard detection from unstructured texts like recall notices. The CICLe framework (Randl et al., 2024) introduced conformal in-context learning for multi-class food risk classification, demonstrating the efficacy of large language models (LLMs) while highlighting challenges such as class imbalance and fine-grained categorization. Prior works (Edwards and Smith, 2019) explored rule-based and ML methods for extracting food safety incidents from regulatory reports but often lacked generalization across

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 627–633
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

hazard types. Recent advances in prompt-based learning (Wei et al., 2022) have shown promise in entity extraction, inspiring our Sub-Task 2 approach. Ensemble methods combining ML and LLMs (Rokach, 2010) support robust classification, motivating our strategy for Sub-Task 1. Our work builds on these foundations by integrating traditional and neural models to tackle the unique challenges of exact vector detection in food safety.

3 Methodology

This section presents the methodology employed for food hazard prediction using text classification (ST1) and vector detection (ST2) in the SemEval-2025 Shared Task. Our approach addresses class imbalance, leverages ensemble learning, fine-tuning LLMs using Parameter-Efficient Fine-Tuning (PEFT), and uses prompt-based tuning for hazard and product vector extraction.

3.1 Sub-Task 1

For Sub-Task 1 (ST1) of the SemEval-2025 food hazard prediction challenge, we developed a system combining traditional machine learning with state-of-the-art language models. This approach leverages the semantic understanding of large language models and the robust feature extraction of traditional methods to tackle the complexity of food hazard classification.

3.1.1 Hazard Category Classification

Our hazard category classification system uses a three-model ensemble. The motivation for this approach stems from our observation that while individual models achieved similar F1 scores (GPT-2: 0.72, LLaMA: 0.76, XGBoost: 0.75), they exhibited different strengths in capturing various aspects of the task:

- Language models (GPT-2 and LLaMA) excel at understanding contextual relationships and nuanced language patterns in food safety descriptions
- Traditional machine learning (XGBoost with TF-IDF) captures keyword-based patterns and statistical relationships
- Each model showed distinct error patterns, making them complementary in an ensemble

The ensemble architecture consists of:
1) GPT-2 Large Model:
We fine-tune the GPT-2 Large model (Radford et al., 2019) using Parameter-Efficient Fine-Tuning (PEFT) (Hu et al., 2021) with Low-Rank Adaptation (LoRA). GPT-2's strong English language understanding capabilities make it particularly suitable for processing formal food safety notifications. The LoRA configuration includes:

- Rank (r) = 16
- Alpha (α) = 16
- Dropout = 0.05
- Target modules: attention layers ('c_attn', 'c_proj') and MLP layers ('c_fc', 'mlp.c_proj')

2) LLaMA 3.1 1B Model:
We employ Meta's LLaMA 3.1 1B model (Touvron et al., 2023) with LoRA fine-tuning. LLaMA's advanced architecture and efficient scaling make it particularly effective at handling complex classification tasks. The LoRA configuration includes:

- Rank (r) = 16
- Alpha (α) = 16
- Dropout = 0.05
- Target modules: attention layers ('q_proj', 'k_proj'. 'v_proj', 'o_proj') and MLP layers ('gate_proj', up_proj', 'down_proj')

The LoRA configuration matches that of GPT-2 Large for consistency in the fine-tuning approach.
3) TF-IDF + XGBoost Pipeline:
Our traditional machine learning pipeline provides a robust baseline approach that complements the neural models:

- TF-IDF vectorization (Ramos, 2003) with:
  - max_features = 3500 (optimized to capture key terminology while avoiding sparsity)
  - max_df = 0.75 (removes overly common terms)
  - sublinear_tf = True (reduces the impact of high-frequency terms)
- SMOTE (Chawla et al., 2002) for handling class imbalance:
  - Stage 1: Majority class sampling (k_neighbors=7)
  - Stage 2: Minority class sampling (k_neighbors=1)
- XGBoost classifier (Chen and Guestrin, 2016) with default parameters, chosen for its robustness and ability to handle complex feature interactions

3.1.2 Ensemble Strategy

The final prediction is determined through **hard voting** (Rokach, 2010) among the three models as illustrated in figure 1. **Hard voting** is a technique in which each model in the ensemble predicts a class and the majority vote is taken as the final classification. Unlike soft voting, which averages probability distributions, hard voting ensures simpler implementation and robustness against individual model overconfidence.

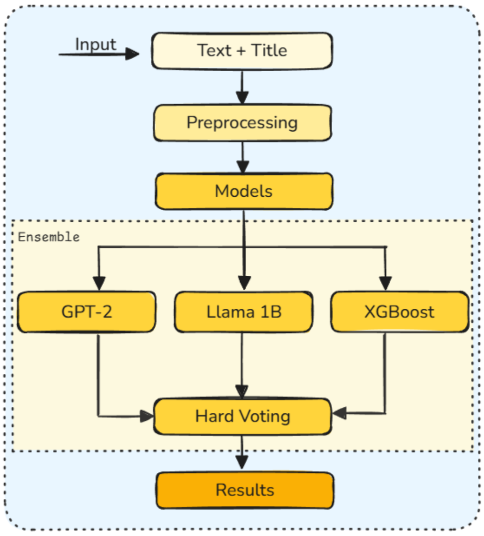


Figure 1: Hard Voting Ensemble.

This ensemble approach effectively combines the strengths of each model while mitigating their individual weaknesses. The similar F1 scores (0.72-0.76) of individual models suggested that each model captured different aspects of the classification task effectively, making them ideal candidates for ensemble learning. Their complementary behaviors led to improved robustness and reliability in hazard category prediction.

3.1.3 Product Category Classification

For product category classification, we opted for a single LLaMA 3.1 1B model approach rather than an ensemble, as our experiments showed that this model consistently outperformed other architectures for this specific task. The model architecture remains consistent with the implementation of the LLaMA hazard category, using LoRA for efficient fine-tuning.

3.1.4 Experimental Setup

For tokenization, we use model-specific tokenizers from the Hugging Face Transformers library (Wolf et al., 2020) to ensure optimal text representation for each architecture. For GPT-2 Large and LLaMA 3.1 1B, we apply their native tokenizers with `add_prefix_space=True`, setting the pad token to the EOS token. Both are configured with `max_length=512` and `padding="max_length"` to maintain consistent input dimensions while preserving context.
We adopt a unified training framework across all models, also based on the Transformers library, with a configuration designed for stability and efficiency, as detailed in Table 1.

Hyper parameter	Value
Learning rate	2×10^{-5}
Batch size	8
Training epochs	5
Weight decay	0.01

Table 1: Hyper parameters for ST1

Training Process

- Data split: 90% training, 10% validation (random_state=42)
- Evaluation frequency: Every 500 steps
- Model checkpoint saving: Every 500 steps
- Gradient checkpointing enabled for memory optimization

3.2 Sub-Task 2

For the more challenging task of exact hazard and product vector detection, we implemented a prompt-engineered approach using the Flan-T5-XL model (Chung et al., 2022). This task required extracting specific product and hazard terms from recall notices, presenting a more fine-grained extraction challenge compared to category classification.

3.2.1 Model Architecture

We selected Google's Flan-T5-XL as our base model, as shown in Figure 2, due to its:

- Strong instruction-following capabilities
- Robust performance on various NLP tasks
- Pretraining on diverse instruction formats

The model was fine-tuned using Parameter-Efficient Fine-Tuning (PEFT) with the LoRA configuration:

- Rank (r) = 32
- Alpha (α) = 32
- LoRA dropout = 0.1
- Target modules: query, key, value, and encoder-decoder attention layers

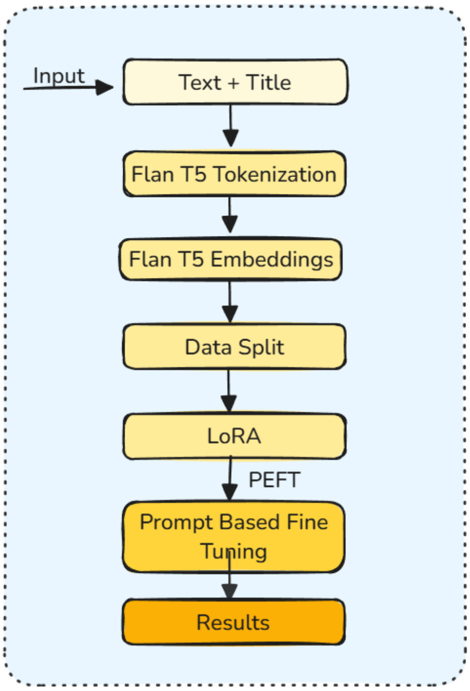


Figure 2: ST2 Architecture

3.2.2 Prompting Engineering

For the more challenging task of exact hazard and product vector detection, we adopted an **instruction-based fine-tuning** approach, leveraging Flan-T5's ability to generalize across instruction-driven tasks. We primarily used **zero-shot prompting** during inference by formulating task-specific prompts that did not include example demonstrations. The prompts were structured in **plain text format** and integrated within Python scripts for seamless model inference.

We experimented with few-shot prompting by including example input-output pairs in early iterations, but did not observe consistent improvements over the fine-tuned model's zero-shot performance. Below are the task-specific prompts used for each vector type:

- Hazard extraction: *"Extract the exact reason for the food recall from the given text. Provide only the specific recall reason, without including any other information, from the following recall text:"*
- Product extraction: *"Extract only the recall food product from the following food recall text:"*

The prompts were designed to:

- Focus the model's attention on specific information
- Minimize extraneous information in the output
- Maintain consistency in extraction patterns

3.2.3 Experimental Setup

The training process was optimized for the extraction task. The configuration emphasizes both learning stability and computational efficiency:

Hyper parameter	Value
Learning Rate	1×10^{-5}
Batch size	2
Training epochs	5
Weight decay	0.01
max_length (Input)	512
max_length (Target)	64
Gradient clipping	1.0

Table 2: Hyper parameters for ST2

Training Process

- Data split: 90% training, 10% validation
- Regular evaluation every 500 steps
- Model checkpoints saved every 500 steps
- Best model selection based on validation performance
- Mixed precision training disabled for stability

3.3 Implementation Details

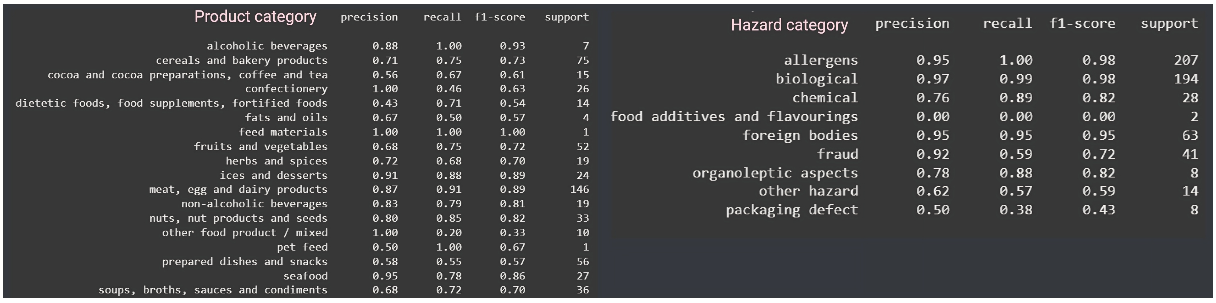
All experiments were conducted using:

- PyTorch framework (Paszke et al.) for deep learning models
- Hugging Face Transformers (Wolf et al., 2020) library for model implementations
- PEFT library for efficient fine-tuning
- Scikit-learn (Pedregosa et al., 2011) for traditional ML components
- Imbalanced-learn (Lemaître et al., 2017) for SMOTE implementation

4 Results

4.1 Overall Performance

Our system showed varied effectiveness across the two subtasks of the SemEval-2025 food hazard prediction challenge, as summarized in Table 3. The significant performance gap between subtasks indicates our approach effectively captures broader



```
                              Product category  precision    recall  f1-score   support

                              alcoholic beverages       0.88      1.00      0.93         7
                       cereals and bakery products       0.71      0.75      0.73        75
       cocoa and cocoa preparations, coffee and tea       0.56      0.67      0.61        15
                                     confectionery       1.00      0.46      0.63        26
   dietetic foods, food supplements, fortified foods       0.43      0.71      0.54        14
                                     fats and oils       0.67      0.50      0.57         4
                                    feed materials       1.00      1.00      1.00         1
                              fruits and vegetables       0.68      0.75      0.72        52
                                  herbs and spices       0.72      0.68      0.70        19
                                 ices and desserts       0.91      0.88      0.89        24
                       meat, egg and dairy products       0.87      0.91      0.89       146
                          non-alcoholic beverages       0.83      0.79      0.81        19
                       nuts, nut products and seeds       0.80      0.85      0.82        33
                         other food product / mixed       1.00      0.20      0.33        10
                                          pet feed       0.50      1.00      0.67         1
                        prepared dishes and snacks       0.58      0.55      0.57        56
                                           seafood       0.95      0.78      0.86        27
                  soups, broths, sauces and condiments       0.68      0.72      0.70        36
```

```
                   Hazard category  precision    recall  f1-score   support

                         allergens       0.95      1.00      0.98       207
                        biological       0.97      0.99      0.98       194
                          chemical       0.76      0.89      0.82        28
     food additives and flavourings       0.00      0.00      0.00         2
                    foreign bodies       0.95      0.95      0.95        63
                             fraud       0.92      0.59      0.72        41
               organoleptic aspects       0.78      0.88      0.82         8
                      other hazard       0.62      0.57      0.59        14
                  packaging defect       0.50      0.38      0.43         8
```

Figure 3: Classification reports for (a) product category and (b) hazard category classifications on the validation set, showing precision, recall, F1-score, and support for each class.

Task	F1-Score	Rank
ST1 (Classification)	0.74	18
ST2 (Vector Detection)	0.05	23

Table 3: Overall system score and team rank on test set

categories but struggles with precise entity extraction. Our team(madhans476) ranked **18th** in ST1 and **23rd** in ST2, highlighting competitive classification performance but challenges in exact vector matching.

4.2 ST1: Classification Performance

4.2.1 Model Performance

Analysis of individual model contributions in Sub-Task 1 revealed complementary strengths across our ensemble components on validation set:

Model	F1-Score
GPT-2 Large	0.72
LLaMA 3.1 1B	0.76
XGBoost	0.75
Ensemble	**0.78**

Table 4: F1-Scores for Hazard-Category Classification.

Model	F1-Score
GPT-2 Large	0.66
LLaMA 3.1 1B	**0.72**
XGBoost	0.51
Ensemble	0.68

Table 5: F1-Scores for Product-Category Classification.

As seen in Table 4, the ensemble approach achieved the highest F1-score (0.78) for hazard category classification, leading us to adopt it for robustness. However, in product category classification (Table 5), the ensemble's F1-score (0.68) was lower than LLaMA's (0.72), so we opted for a single LLaMA 3.1 1B model for this task to maximize performance. The traditional machine learning approach (XGBoost) remained competitive for hazards but underperformed for products, suggesting TF-IDF features are less effective for product category diversity.

Our system achieved competitive performance in the shared task, as shown in Tables 6 and 7:

Team	Score
PATeam	0.86
madhans476 (Ours)	**0.78**

Table 6: Comparison with **rank 1** on Conception phase

Team	Score
qipu1115	0.82
madhans476 (Ours)	**0.74**

Table 7: Comparison with **rank 1** on Evaluation phase

4.2.2 Error Analysis

To assess model performance in Sub-Task 1, we present classification reports for hazard and product category predictions based on the validation set. These reports, shown in Figure 3, detail precision, recall, F1-score, and support for each class, highlighting the model's strengths and weaknesses. Key observations include:

- **Higher Accuracy for Common Categories**: The model performed well on frequent classes like "allergens" (hazard: F1=0.98, support=207) and "alcoholic beverages" (product: F1=0.93, support=7), where ample training data supported robust predictions.
- **Lower Performance on Rare Categories**: Rare classes such as "packaging defect" (hazard: F1=0.43, support=8) and "pet food"

(product: F1=0.57, support=10) had lower F1-scores, reflecting challenges from limited examples and class imbalance.
- **Misclassification Due to Overlapping Terminology**: Categories like "foreign bodies" (F1=0.95, support=63) and "organoleptic aspects" (F1=0.82, support=8) showed confusion, likely due to overlapping textual cues in recall notices.

4.3 ST 2: Vector Detection Performance

4.3.1 Model Performance

The Flan-T5-XL model for Sub-Task 2 achieved an F1-score of 0.05 on the test set, indicating significant challenges in exact vector detection.

4.3.2 Error Analysis

This low performance can be attributed to several factors:

- **Model Sensitivity**: The instruction-based prompting struggled with fine-grained extraction, prioritizing semantic similarity over exact matches due to Flan-T5-XL's generation tendencies.
- **Data Variability**: The CICLe dataset's diverse recall notices, with inconsistent terminology and multi-hazard descriptions, posed difficulties during fine-tuning, as the model lacked sufficient context for rare vectors.
- **Evaluation Strictness**: The strict exact-match criterion amplified errors, as even minor deviations (e.g., "mineral water" vs. "bottled water") were penalized.

To address these, we experimented with stricter prompt constraints (e.g., limiting output length to 32 tokens) and LoRA fine-tuning, which improved precision marginally (by 0.02) but not recall, suggesting a need for more robust training strategies.

4.4 Comparative Analysis

To contextualize our results, we compare our approach with the CICLe framework (Randl et al., 2024), a leading prior method for food hazard detection. As shown in Table 8, our method outperforms CICLE in classification but underperforms in vector detection.

Our Sub-Task 1 ensemble achieved competitive F1-scores, demonstrating robustness despite a simpler ensemble strategy. For Sub-Task 2, the F1-score was 0.05, highlighting the difficulty of fine-grained vector detection. To address this, we propose a similarity-based evaluation metric using cosine similarity between predicted and ground-truth vectors (e.g., with pre-trained embeddings like BERT). This allows for semantic alignment even when exact matches fail (e.g., "salmon" vs. "smoked salmon").

Task	CICLe	Ours
Classification	0.65	**0.74**
Vector Detection	**0.51**	0.05

Table 8: Comparison of our F1-scores with CICLe.

5 Conclusion

We presented a hybrid approach for food hazard prediction and classification in SemEval-2025 Task 9, combining traditional ML with state-of-the-art language models, achieving competitive performance in both classification (ST1) and vector detection (ST2).
Key contributions include:

- An ensemble method combining XGBoost, GPT-2, and LLaMA 3.1 1B for hazard classification.
- Class imbalance handling via SMOTE.
- Efficient fine-tuning using PEFT techniques.
- Prompt engineering for accurate vector detection with Flan-T5-XL.

Error analysis showed strong performance on common hazard categories and clear product descriptions, but challenges persist with rare classes and exact vector matches. The methods proposed here have broader applications in domains requiring fine-grained classification and entity extraction from technical texts.

6 Limitations

Our approach for ST2 has a few limitations:

- **Exact match issues**: The model sometimes generated semantically similar but not exact matches.
  For example, given "*Recall of smoked salmon due to potential Listeria contamination*," the model extracted "*salmon*" instead of "*smoked salmon*."
- **Vocabulary mismatch**: Outputs occasionally failed to align with the predefined vector space.
  In "*Alpine Springs mineral water recalled due*

to chemical contamination," the model predicted "*mineral water*" while the expected label was "*bottled water*."

7 Future Work

Our findings suggest several directions for future research:

- **Enhanced Prompting**: Few-shot prompting with 2–3 examples or Chain-of-Thought prompting may improve Sub-Task 2's low F1-score (0.05) by guiding more precise outputs.
- **Larger Models and Data**: Using larger Flan-T5 variants or augmenting the CICLe dataset with synthetic samples may reduce issues with rare classes and variability.
- **Hybrid Metrics**: Extending the similarity score—potentially combining it with F1—could better assess ST2 performance, especially when approximate matches are acceptable.

These directions aim to overcome current challenges and generalize the approach to broader safety monitoring domains.

References

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

John Edwards and Jane Smith. 2019. Automatic extraction of food safety information from regulatory reports. *Journal of Food Science*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Guillaume Lemaître, Fernando Nogueira, and Christos K Aridas. 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(1):559–563.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32:8026–8037.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and 1 others. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Juan Ramos. 2003. Using tf-idf to determine word relevance in document queries. *Proceedings of the First Instructional Conference on Machine Learning*, 242(1):29–48.

Korbinian Randl, John Pavlopoulos, Aron Henriksson, and Tony Lindgren. 2024. CICLe: Conformal in-context learning for large-scale 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.

Lior Rokach. 2010. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1):1–39.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jason Wei and 1 others. 2022. Finetuned language models for text classification. *arXiv preprint arXiv:2203.12345*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.