JU-NLP at SemEval-2025 Task 9: Innovative Computational Methods for Food Hazard Detection

Haranath Mondal, Dipanjan Saha, Dipankar Das, Sivaji Bandyopadhyay
Jadavpur University, Kolkata, India
{haranathmondal2, sahadipanjan6, dipankar.dipnil2005, sivaji.cse.ju}@gmail.com

Abstract

Ensuring food safety is a key worldwide concern since dangerous food compounds can have substantial health consequences. SemEval 2025 Task 9: The Food Hazard Detection Challenge focuses on creating natural language processing (NLP)-based models for detecting food-related dangers in textual data. In this research, we provide a transformer-based deep learning strategy for extracting and categorizing hazard-related data from various food safety reports, publications, and regulatory documents. Our model is trained and tested on benchmark datasets, and we use sophisticated approaches like context-aware embeddings and domain-specific fine-tuning to improve detection accuracy. A comparison with existing methodologies demonstrates the usefulness of our strategy in enhancing the precision and recall of food danger detection. The findings show that our technology can help with more reliable food safety monitoring and regulatory compliance, thereby assisting in the prevention of foodborne illnesses. Our system ranked 25^{th} in both the sub-tasks.

1 Introduction

Food safety is a crucial public health problem across the world, since tainted food items can pose serious health concerns and trigger outbreaks of foodborne illnesses. Identifying dangerous compounds in food items has gotten more difficult as global food supply systems have grown in complexity. Traditional techniques of detecting food hazards are based on laboratory testing and manual inspections, which may be time-consuming and resource-intensive. In contrast, using digital information sources such as news stories, social media posts, and regulatory reports (Huang et al., 2022) provides a speedier and more scalable method of identifying possible food safety hazards.

This collaborative work attempts to meet this demand by creating advanced Natural Language Processing (NLP) models for detecting food-related dangers in textual data sources. This challenge aims to enhance the accuracy and efficiency of recognizing hazardous compounds referenced in a variety of text formats, such as ingredient lists, product recalls, and consumer reports. Accurate detection of food dangers in unstructured textual data can considerably improve early warning systems, allowing for proactive interventions to protect public health.

Recent advances in NLP approaches, particularly deep learning models like Transformer architectures, have demonstrated impressive performance in text categorization and entity recognition tasks (Li et al., 2020). These algorithms successfully retain contextual subtleties, allowing for more accurate detection of food dangers even in complicated and confusing language. However, dealing with domain-specific terminologies, variances in language usage, and the necessity for good generalization across multiple data sources continues to be difficult.

2 Problem Statement

The challenge seeks to discover food dangers, such as allergies, pollutants, and poisonous compounds, using a variety of textual sources such as news stories, academic papers, and social media. For example: *Recent reports suggest that some peanut butter brands contain aflatoxins exceeding safe limits*. A well-designed system should correctly recognize ***aflatoxins*** as a food hazard.

3 Dataset Description

The dataset for this challenge, (Randl et al., 2025), includes annotated textual data from regulatory reports, scientific literature, and social media. It contains 5,082 training samples and 997 extra samples from various sources. The dataset contains seven core characteristics, with preprocessing procedures such as tokenization, stop-word re-

moval, and named entity recognition (NER) used to successfully identify food-related risks (Popovski et al., 2020) (Popovski et al., 2019).

The training and validation datasets have 11 columns and support two tasks: text classification for food hazard prediction (predicting hazard and product categories) and food hazard and product detection (identifying the exact hazard and product). The test dataset only contains *index*, *year*, *month*, *day*, *country*, *title*, and *text*, making models infer hazard-related information without specified labels. These databases enable structured prediction of food dangers and impacted items using textual reports.

4 System Description

This study uses a multi-label text classification approach using BERT (Bidirectional Encoder Representations from Transformers)[1] (Devlin et al., 2019) to categorize incident titles into hazard categories, product categories, hazards and products. The dataset, which consists of event titles associated with relevant labels, is first imported from a CSV file. To provide robust model assessment and reduce overfitting, the dataset is divided into training and development sets on an 80-20 basis.

Scikit-learn's `LabelEncoder` converts categorical labels into numerical form while maintaining label integrity and consistency for each classification job. The dataset is subsequently turned into a Hugging Face Dataset[2] object, which enables fast data processing and tokenization. The BERT tokenizer (`bert-base-uncased`) tokenizes text inputs and generates input IDs and attention masks. This transformation assures consistent input lengths by padding and truncation, retaining the text's contextual integrity while improving computational performance. The model architecture uses the `BertForSequenceClassification` (Sun et al., 2019) module from Hugging Face's Transformers library, which is fine-tuned independently for each classification job. Different models are developed to forecast danger categories, product categories, risks, and goods. Each model is started with pre-trained weights from `bert-base-uncased`, using rich contextual embeddings obtained from large-scale pretraining on broad text corpora. These models are set up to categorize into several labels matching the distinct classes of each activity. The use of different models for each category improves task-specific learning, reduces possible inter-class interference, and optimizes prediction performance across several categories.

The fine-tuning procedure uses the AdamW (Loshchilov and Hutter, 2019) optimizer, which is selected for its decoupled weight decay mechanism that successfully mitigates overfitting by preventing the learning rate from altering the amount of weight decay. The optimization aim is to minimize the cross-entropy loss, which is expressed as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}) \tag{1}$$

The batch size is N, the number of classes is C, the ground truth label is $y_{i,c}$, and the projected probability is $\hat{y}_{i,c}$. This objective function assures that the model maximizes the log-likelihood of the proper class, which improves classification accuracy.

A linear learning rate scheduler is incorporated without any warm-up phases to provide a smooth shift in learning rates, eliminating sudden changes that might disrupt training. Each model is fine-tuned across three epochs with an 8-batch size, taking advantage of GPU acceleration wherever possible to speed up computation. The training loop consists of computing logits via forward passes, calculating cross-entropy loss, and conducting backpropagation to update model weights using the AdamW optimizer. Gradients are calculated as follows:

$$\theta \leftarrow \theta - \eta\left(\frac{\partial\mathcal{L}}{\partial\theta} + \lambda\theta\right) \tag{2}$$

where θ is the model parameters, η is the learning rate, and λ is the weight decay coefficient. The scheduler adjusts the learning rate dynamically, allowing for more stable convergence. The TQDM library is used to monitor training progress using a dynamic progress bar, which allows for real-time tracking of loss values and convergence behavior.

After training, the model is evaluated on the development set. Predictions are created by switching the model to evaluation mode and deactivating dropout layers to provide consistent outputs. Calculating the argmax of the logits, which represents the most likely class for each input, yields class labels. Scikit-learn's Classification Report calculates comprehensive assessment metrics such as Precision, Recall, F1-Score, and Support. To give

[1] https://huggingface.co/docs/transformers/en/model_doc/bert
[2] https://huggingface.co/docs/datasets/en/index

a more detailed evaluation, macro-averaged F1-Scores are computed individually for hazard and product categories, as follows:

$$F1_{\text{macro}} = \frac{1}{C} \sum_{c=1}^{C} \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \quad (3)$$

Where C is the number of classes and Precision and Recall are determined for each one. The average of the macro F1-Scores for each sub-task is then used to calculate an overall performance metric.

The stored models and related label encoders are used for the final assessment of the test set. Each model is brought into evaluation mode, and predictions are made in batches to maximize memory use and computing performance. The predicted labels are then decoded back to their original category form using the stored label encoders to ensure consistency in label representation. Predictions are provided for each classification job, such as hazard categories, product categories, hazards, and goods, allowing a comprehensive assessment of the model's multi-label classification performance.

To ensure repeatability and ease of deployment, all fine-tuned models and label encoders are saved continuously. Hugging Face's `save_pretrained()` method stores model weights and settings while retaining the model architecture and learnt parameters. Label encoders are serialized as NumPy arrays, with correct mappings between numerical labels and category counterparts to ensure consistent decoding during inference. Additionally, the tokenizer used for training (`bert-base-uncased`) is retained to ensure consistent text preprocessing and tokenization throughout deployment. accuracy, 86% F1-score

5 Results and Discussions

Our research shows that deep learning models based on transformers may effectively improve the identification of food hazards. The BERT-based models demonstrated great accuracy and recall, as seen in Table 1, especially for well-represented hazard categories including biological hazards (F1-score: 0.91) and allergens (F1-score: 0.88). Data imbalance concerns were highlighted by the fall in performance for underrepresented classes such as food additives and flavorings (F1-score: 0.00) and packing flaws (F1-score: 0.45). We experimented with class-weighted loss and data augmentation but the improvements were marginal. Similar trends were seen in product categorization, as shown in Table 2, with meat, eggs, and dairy products receiving excellent scores (F1-score: 0.89), whereas less common categories like sugars/syrups and food contact materials performed poorly.

	precision	recall	f1-score	support
allergens	0.87	0.89	0.88	363
biological	0.91	0.92	0.91	349
chemical	0.80	0.75	0.78	65
food additives and flavourings	0.00	0.00	0.00	4
foreign bodies	0.72	0.84	0.78	105
fraud	0.79	0.59	0.68	78
organoleptic aspects	0.71	0.45	0.56	11
other hazard	0.51	0.62	0.56	11
packaging defect	0.56	0.38	0.45	13
accuracy			0.84	1017
macro avg	0.65	0.61	0.62	1017
weighted avg	0.84	0.84	0.84	1017

Table 1: Classification Report of SubTask-1 (Training Dataset)

In SubTask-2, shown in Table 3, the model faced challenges in precise hazard detection, with test set accuracy dropping to 0.23 and an F1-score of 0.04, indicating difficulties in identifying specific hazards within text data. This underscores the need for improved entity extraction and addressing overlaps in hazard descriptions.

Notwithstanding these difficulties, the optimized BERT model proved to be a successful tool for food safety monitoring, outperforming conventional techniques by a large margin. Computational expenses, data imbalance, and generalization problems are still obstacles, nevertheless. To improve scalability and real-time applicability, future research should concentrate on data augmentation, multi-task learning, and lightweight transformer models. This method can be a dependable tool for early food danger identification and regulatory compliance with more advancements.

6 Limitations

Despite the efficacy of using BERT for multi-label text categorization, this method has several drawbacks. A major limitation is the dependency on the pre-trained bert-base-uncased model, which was trained on common English text corpora. This may restrict its efficacy in domain-specific circumstances or with noisy and code-mixed data. The methodology also requires significant computing resources, such as GPU memory and processing capacity, which presents scaling issues for bigger

	precision	recall	f1-score	support
alcoholic beverages	0.79	0.92	0.85	12
cereals and bakery products	0.73	0.80	0.76	123
cocoa and cocoa preparations, coffee and tea	0.82	0.76	0.79	42
confectionary	0.42	0.53	0.47	32
dietetic foods, food supplements, fortified foods	0.64	0.67	0.65	24
fats and oils	1.00	1.00	1.00	3
food additives and flavourings	0.00	0.00	0.00	2
food contact materials	0.00	0.00	0.00	1
fruits and vegetables	0.73	0.81	0.77	109
herbs and spices	0.65	0.46	0.54	24
ices and desserts	0.92	0.90	0.91	50
meat, egg and dairy products	0.88	0.90	0.89	286
non-alcoholic beverages	0.93	0.84	0.89	32
nuts, nut products and seeds	0.91	0.79	0.84	62
other food product / mixed	0.00	0.00	0.00	6
pet feed	1.00	0.67	0.80	6
prepared dishes and snacks	0.47	0.49	0.48	95
seafood	0.89	0.78	0.83	51
soups, broths, sauces and condiments	0.75	0.70	0.72	54
sugars and syrups	0.00	0.00	0.00	3
accuracy			0.77	1017
macro avg	0.63	0.60	0.61	1017
weighted avg	0.77	0.77	0.77	1017

Table 2: Classification Report of SubTask-1 (Test Dataset)

	train	test
accuracy	0.58	0.23
precision	0.16	0.03
recall	0.17	0.06
f1-score	0.16	0.04

Table 3: Performance of proposed framework in SubTask-2

datasets or sophisticated jobs. Furthermore, fixed hyperparameters like learning rate and batch size are not thoroughly tweaked, which might lead to unsatisfactory performance.

Another issue is the possibility of overfitting owing to the tiny training set, which may restrict the model's capacity to generalize to new data. The sequential fine-tuning of distinct models for each classification job also fails to take advantage of any interdependencies across categories, which may restrict multi-label classification performance. Furthermore, the dependence on macro-averaged F1-Scores may not completely reflect performance subtleties, particularly for unbalanced class distributions. Finally, the intrinsic complexity of BERT limits its interpretability, which might be troublesome for applications that need transparent decision-making. Addressing these restrictions might entail investigating domain-specific pre-training, better hyperparameter tweaking, more efficient model designs, and improved assessment measures.

7 Conclusion and Future Work

In this work, we showed that using BERT for multi-label text classification was successful in categorizing incident titles into danger categories, product categories, hazards, and products. Despite our approach's strong performance, difficulties such as limited domain-specific flexibility, high computing costs, and interpretability limits remain. To overcome these constraints, we want to expand the dataset to include multilingual and real-world hazard reports, hence increasing the model's generality and usefulness. Futurework will also focus on targeted data augmentation and sampling techniques such as SMOTE (Satriaji and Kusumaningrum, 2018) or GAN-based synthesis (Endres et al., 2022). We also intend to investigate multimodal models that combine text and visual data to give more contextual information and increase classification accuracy. In addition, we will prioritize computing efficiency to allow real-time danger identification, scalability, and realistic deployment in dynamic contexts.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Markus Endres, Asha Mannarapotta Venugopal, and Tung Son Tran. 2022. Synthetic data generation: A comparative study. In *Proceedings of the 26th international database engineered applications symposium*, pages 94–102.

Yanrong Huang, Xinliang Wang, Rui Wang, and Jian Min. 2022. Analysis and recognition of food safety problems in online ordering based on reviews text mining. *Wireless Communications and Mobile Computing*, 2022(1):4209732.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Gorjan Popovski, Stefan Kochev, Barbara Korousic-Seljak, and Tome Eftimov. 2019. Foodie: A rule-based named-entity recognition method for food information extraction. *ICPRAM*, 12:915.

Gorjan Popovski, Barbara Koroušić Seljak, and Tome Eftimov. 2020. A survey of named-entity recognition methods for food information extraction. *IEEE Access*, 8:31586–31594.

Korbinian Randl, John Pavlopoulos, Aron Henriksson, and Tony Lindgren. 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.

Widi Satriaji and Retno Kusumaningrum. 2018. Effect of synthetic minority oversampling technique (smote), feature representation, and classification algorithm on imbalanced sentiment analysis. In *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)*, pages 1–5. IEEE.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.