Fossils at SemEval-2025 Task 9: Tasting Loss Functions for Food Hazard Detection in Text Reports

Aman Sinha[1] **and Federica Gamba**[2]
[1]Institut Elie Cartan de Lorraine, Université de Lorraine, Nancy, France
[2]Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic

Abstract

Food hazard detection is an emerging field where NLP solutions are being explored. Despite the recent accessibility of powerful language models, one of the key challenges that still persists is the high class imbalance within datasets, often referred to in the literature as the *long tail problem*. In this work, we present a study exploring different loss functions borrowed from the field of visual recognition, to tackle long-tailed class imbalance for food hazard detection in text reports. Our submission to SemEval-2025 Task 9 on the Food Hazard Detection Challenge shows how re-weighting mechanisms in loss functions prove beneficial in class imbalance scenarios. In particular, we empirically show that class-balanced and focal loss functions outperform all other loss strategies for Subtask 1 and 2 respectively.

Fossils-FHD

1 Introduction

Ensuring food safety is a critical global challenge, as contaminated food can lead to severe health risks and economic losses. Contaminants such as biological hazards (e.g., Salmonella, Listeria, E. coli), chemical hazards (e.g., pesticide residues, heavy metals, food additives), and physical hazards (e.g., glass, plastic, metal fragments) pose significant risks to consumers. Early detection of such hazards is thus essential for protecting public health. With the growing availability of text-based data sources, such as news articles and social media posts, Natural Language Processing (NLP) techniques can represent an asset to provide scalable solutions for detecting and classifying food hazards from unstructured text.

This observation has motivated our participation in the SemEval-2025 Task 9 (Randl et al., 2025), which focuses on the "Food Hazard Detection Challenge" and aims at developing classification systems for titles of food-incident reports collected from the web. The provided dataset includes manually labeled English food recall titles from official food agency websites (e.g., FDA). The task comprises of two sub-tasks: (a) **Subtask1**: *Text classification for food hazard prediction*, predicting the type of hazard and product; (b) **Subtask2**: *Food hazard and product "vector" detection*, predicting the exact hazard and product.

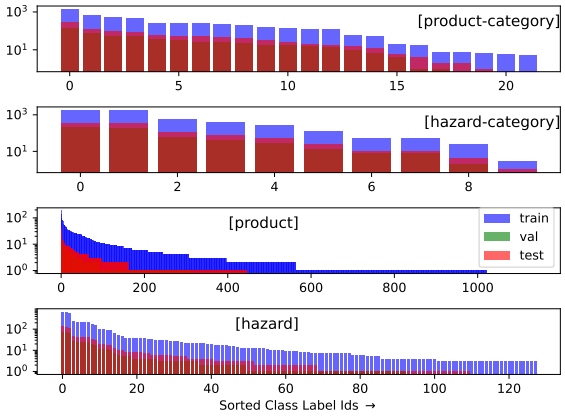


Figure 1: The *long tail* problem in the food hazard detection dataset. The plots shows overlapping bar plots for train-val-test splits with x-axis denoting label ids (descending sorted by frequency for each split) and y-axis as log-scaled class frequency.

Despite the potential of NLP techniques for food safety monitoring, several challenges still remain. In particular, we observe that the dataset (see fig. 1) suffers from a *long tail* problem (Zhang et al., 2023), showing a substantial aggregate frequency for classes that individually have very low frequency. In addition to this, at times, the test set may include previously unseen classes that are absent from the training data[1]. Therefore, this scenario presents two distinct challenges: adapting the model to classes for which it has encountered (i) a very low number of instances, and (ii) no instances

[1]This applies to the *product* set of labels. (See fig. 2)

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 1515–1521

at all. Moreover, when the model is trained on the training set, it inherently assumes a similar class distribution in the test set. This reflects the independent and identically distributed (IID) assumption, which presumes that the training and test datasets share the same underlying distribution.

The long-tailed class imbalance represents a common problem in practical visual recognition tasks, often limiting the practicality of deep network based recognition models in real-world applications, since they can be easily biased towards dominant classes and perform poorly on tail classes. Acknowledging a similar issue in the setting of the Food Hazard Detection in text reports, in this paper we present the following contributions: (a) We participate in both subtasks of SemEval-2025 Task 9 on the Food Hazard Detection Challenge. (b) We direct our investigation toward assessing the effectiveness of various loss functions in mitigating the impact of long-tailed class imbalance. (c) Our final submission for each subtask is an ensemble of multiple models trained using different loss functions.

2 Related Work

Food Hazard Detection in Text Reports To date, NLP research on food hazards has primarily been framed as a binary detection task, focusing on detecting the presence or absence of hazards rather than classifying incidents into specific hazard categories. This approach has been used to generate warnings from online texts, for instance, by Maharana et al. (2019), who leveraged Amazon reviews matched with FDA food recall announcements for hazard detection, and by Tao et al. (2023), who combined Twitter data with reports from the U.S. Centers for Disease Control and Prevention. As existing datasets (Hu et al., 2022) follow the same approach and focus on hazard detection, Randl et al. (2024) introduced the first openly accessible resource for text classification of food hazards. The dataset is structured into two levels of granularity and provides the data for SemEval-2025 Task 9 (Randl et al., 2025).

Long Tail Imbalance in NLP Severe class imbalance is one of the most prominent challenges that Randl et al. (2024) identified in the dataset, affecting overall classification performance in particular for low-frequency categories.

In general, various approaches have been explored to address long-tail imbalance. Among those, data augmentation techniques aim to artificially increase the number of samples in low-frequency classes, by generating synthetic examples for underrepresented categories (Wei and Zou, 2019; Anaby-Tavor et al., 2020).

Other approaches include oversampling and undersampling techniques. In Random Under-Sampling (RUS), a subset of head-class samples is randomly selected while the remaining samples are discarded to match the number of tail-class instances. Random Over-Sampling (ROS) randomly reproduces tail-class samples to match the number of head-class samples. To tackle overfitting that often results from ROS, Synthetic Minority Over-Sampling Technique (SMOTE) (Chawla et al., 2002) creates a new artificial sample through interpolation between each existing head-class sample. Furthermore, transfer learning can enhance model performance by leveraging knowledge from head classes to improve tail-class representations (Wang et al., 2017), while decoupled learning (Kang et al., 2020) divides learning into two stages: (a) applying end-to-end learning using conventional methods for representation learning, and (b) fixing the feature extractor while retraining the downstream task model for classification.

Another strategy that has been explored is cost-sensitive learning, an algorithm-level approach that assigns higher weights to underrepresented classes to mitigate bias toward frequent categories and improve model generalization. This strategy is often adopted to tackle the long-tail problem in the visual recognition field (Zhang et al., 2023). Unlike more complex solutions that often require extensive resources, loss function modifications offer a lightweight and interpretable strategy that can enhance model performance even with limited data. As a fundamental aspect of model training, they do not replace but rather complement advanced techniques, making them broadly applicable across different models. In light of this, we chose to investigate loss modification as the primary strategy for the shared task.

3 Theoretical Background

3.1 Problem Formulation

We consider a supervised learning setting with N training samples denoted by pair $<X_i, y_i>$ using which we want to train a classifier $f(\theta)$ for a task T which has C train classes such that,

$$f_T(X_{unseen}|\theta) = \hat{p}_{unseen}$$

Here, p_{unseen} is predicted probability vector of length C, where j_{th} element of the vector corresponds to the predicted probability for X_{unseen} associated with class j.

3.2 Loss Functions

In line with the above setting, we describe below five loss functions that we borrow from the literature, in comparison to the standard cross-entropy ($L_{\texttt{ce}}$) loss function:

Weighted CrossEntropy Loss is denoted by $L_{\texttt{wce}}$ and is given by:

$$L_{\texttt{wce}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} \mathbf{w}_j y_{ij} \texttt{log}(\hat{p}_{ij}) \quad (1)$$

where $\mathbf{w}_j$ is the weight for class j and class weights $\mathbf{w}$ are provided to handle class imbalance.

Focal Loss denoted by $L_{\texttt{fl}}$ is another enhancement of the standard cross-entropy loss designed to address class imbalance (Lin et al., 2017) by focusing on "hard" examples, while reducing the loss for "easy" examples. For a multi-class classification problem, it is defined as:

$$L_{\texttt{fl}} = -\alpha_t(1-\hat{p}_{i,y_i})^{\gamma}\texttt{log}(\hat{p}_{i,y_i}) \quad (2)$$

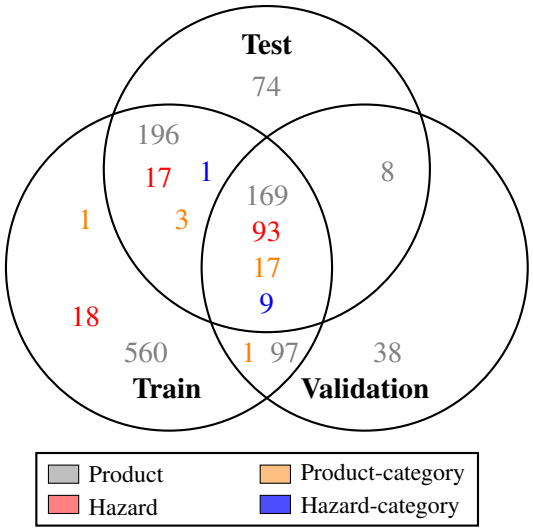
where γ is the focusing parameter to reduce the contribution of "easy" examples (default set to 2.0).

Class-Balanced Loss is designed to address the issue of class imbalance by reweighting the loss contribution of each class based on its effective number of samples. The Class-Balanced Loss (Cui et al., 2019), denoted by $L_{\texttt{cb}}$, uses the following weight for each class, denoted as follows:

$$\mathbf{w}_c = \frac{1-\beta}{1-\beta^{n_c}} \quad (3)$$

where, n_c is the number of samples in class c; β is a hyperparameter ($0 \leq \beta < 1$) controlling the effect of reweighting. The final loss is scaled as: $\mathbf{w}_c$ x $L_{\texttt{ce}}$ loss.

Equalization Loss is used primarily in object detection tasks to address the class imbalance problem, especially for cases with a heavy foreground-background imbalance or long-tailed distributions (Tan et al., 2020). It modifies the standard cross-entropy loss with a suppression term for negative samples based on their class frequencies. It aims to balance the gradients from positive and negative samples.



	Product-cat.	Hazard-cat.	Product	Hazard
Train	22	10	1022	128
Validation	18	9	312	93
Test	20	10	447	110

Figure 2: Entity distribution and overlap across dataset splits. The Venn diagram shows the overlap of unique entity types among the train, validation, and test sets. Colored counts indicate shared entities across splits. The table summarizes the total count of each entity type per split for the four tasks.

$$L_{\texttt{eq}} = -\sum_{i=1}^{C} y_i log(p_i) - (1-y_i) * w_i * log(1-p_i) \quad (4)$$

where y_i is ground truth one-hot vector; p_i is predicted probability for class i; and w_i is the suppression weight for negative samples, often defined based on class frequencies or other heuristics.

LDAM Loss short for Label-Distribution-Aware Margin Loss ($L_{\texttt{ldam}}$) tackles the issue of long tail class imbalance (Cao et al., 2019) by adjusting the decision boundary for classes based on their frequency by adding a margin, which is inversely proportional to the square root of class frequencies.

$$L_{\texttt{ldam}} = -\frac{1}{N}\sum_{i=1}^{N} \texttt{log}\left(\frac{exp(z_{y_i} - \Delta_{y_i})}{\sum_{j=1}^{C} exp(z_j)}\right) \quad (5)$$

where, z_j is logits for class j, y_i is the ground truth class for sample i, N is the total number of samples. Δ_c is modified margin for class c and is defined as $\Delta_c = \frac{C}{\sqrt{N_c}}$. where C is a hyperparameter controlling the scale of the margin and N_c is the number of samples for class c.

	Validation				Test			
	product-cat.	**hazard-cat.**	**product**	**hazard**	**product-cat.**	**hazard-cat.**	**product**	**hazard**
$L_{\texttt{cb}}$	$\mathbf{68.66}_{3.9}$	$\mathbf{80.99}_{4.5}$	$0.00_{0.0}$	$54.57_{15.2}$	$\mathbf{69.93}_{3.9}$	$75.01_{2.0}$	$0.00_{0.0}$	$55.28_{15.5}$
$L_{\texttt{eq}}$	$64.95_{2.1}$	$78.16_{5.5}$	$24.63_{4.3}$	$59.25_{3.5}$	$64.65_{3.3}$	$75.38_{3.1}$	$24.85_{5.9}$	$56.86_{3.6}$
$L_{\texttt{f1}}$	$65.41_{2.0}$	$78.41_{5.4}$	$\mathbf{25.35}_{4.5}$	$\mathbf{60.98}_{3.2}$	$66.74_{3.7}$	$75.07_{1.9}$	$\mathbf{25.78}_{6.0}$	$\mathbf{60.30}_{3.4}$
$L_{\texttt{ldam}}$	$66.68_{5.6}$	$78.85_{5.2}$	$0.00_{0.0}$	$46.17_{25.5}$	$66.11_{5.4}$	$72.99_{3.5}$	$0.00_{0.0}$	$44.88_{24.8}$
$L_{\texttt{wce}}$	$65.95_{5.0}$	$77.88_{11.8}$	$0.00_{0.0}$	$54.74_{16.2}$	$66.32_{4.4}$	$71.13_{10.5}$	$0.00_{0.0}$	$52.84_{15.6}$
$L_{\texttt{ce}}$	$64.71_{2.6}$	$78.13_{5.7}$	$24.60_{4.6}$	$59.54_{2.7}$	$64.42_{3.1}$	$\mathbf{75.94}_{3.0}$	$25.00_{6.0}$	$58.34_{3.3}$

Table 1: Overall results on individual tasks. Each score is the mean of all the configuration with 3 seed runs, while the subscript depicts standard deviation. **Bold** denotes the highest overall score across all the loss functions.

4 Experimental Setup

Dataset Description. The dataset contains timestamp (dd/mm/yyyy) (`timestamp`), title (`title`) and associated text (`text`) along with four labels for every sample. These labels include the product/ hazard category and product/ hazard name. Overall, the dataset comprises of 5082 training samples, 565 validation samples and 997 test samples. The training set in total contains 22 unique labels for *product-category*, 10 for *hazard-category* which together is part of **Subtask1**; further, it contains 1022 unqiue labels for *product* and 128 for *hazard* labels as a part of **Subtask2**. The distribution of unique labels in different data splits is shown in the Venn diagram in fig. 2.

Evaluation Metrics. The two subtasks are evaluated separately by averaging over the macro-F1 ($F1$) over hazard- and product- related tasks.

$$\textbf{Subtask1} = \frac{F1_{\text{product-category}} + F1_{\text{hazard-category}}}{2}$$

$$\textbf{Subtask2} = \frac{F1_{\text{product}} + F1_{\text{hazard}}}{2}$$

PLMs. We use `bert-base-cased` as our pretrained language model (PLM) to perform experiments investigating the impact of different types of loss function.

Hyperparameters. We perform hyperparameter search with different configurations for each loss function. We utilize different learning rates (lr), batch sizes (bs), inputs (I) and run each configuration for 50 epochs with early stopping using 3 different seeds. Details are provided in the Appendix A.

5 Results

In Table 1, we present the overall mean and standard deviation across all configurations we experimented with, using validation data as our test set. The results are reported for different loss functions in comparison to the cross-entropy loss function ($L_{\texttt{ce}}$).

	Validation		Test	
	Subtask1	**Subtask2**	**Subtask1**	**Subtask2**
$L_{\texttt{cb}}$	$\mathbf{74.32}_{2.9}$	$27.29_{7.6}$	$\mathbf{72.59}_{2.4}$	$27.64_{7.7}$
$L_{\texttt{eq}}$	$73.33_{2.8}$	$42.58_{3.3}$	$69.90_{1.9}$	$41.29_{4.1}$
$L_{\texttt{f1}}$	$73.88_{3.1}$	$\mathbf{43.79}_{3.2}$	$70.86_{2.3}$	$\mathbf{43.44}_{4.3}$
$L_{\texttt{ldam}}$	$72.78_{4.8}$	$23.08_{1.3}$	$69.77_{4.1}$	$23.31_{11.7}$
$L_{\texttt{wce}}$	$72.12_{6.8}$	$27.37_{8.1}$	$68.78_{6.5}$	$26.42_{7.8}$
$L_{\texttt{ce}}$	$73.36_{3.8}$	$42.59_{3.2}$	$70.13_{2.3}$	$42.17_{4.03}$

Table 2: Overall results for both the subtasks. **Bold** denotes the highest avg. score across all the loss functions.

We performed initial experiments investigating the use of only-`title`, only-`text`, and `title+text`. We obtained the best results with `title+text` and continued the rest of the experiments using `title+text` as input.

We observe that on the validation set, for *product-category* and *hazard-category* class-balanced ($L_{\texttt{cb}}$) outperforms all the other loss function strategies. Further, for *product* and *hazard*, focal loss ($L_{\texttt{f1}}$) outperforms all other loss functions. This trend is followed on the test set as well except in the case of *hazard-category*, where cross-entropy loss function ($L_{\texttt{ce}}$) outperforms class-balanced loss function ($L_{\texttt{cb}}$).

However, when subjected to the evaluation metrics provided by the SemEval-2025 Task 9 (Randl et al., 2025), we observe a consistent trend where the class-balanced loss function ($L_{\texttt{cb}}$) and the focal loss function ($L_{\texttt{f1}}$) outperform all other loss function strategies for **Subtask 1** and **Subtask 2**, respectively.

Our final submission. We prepare an ensemble of multiple configurations using various loss func-

	Subtask1				Subtask2		
	lr	bs	score		lr	bs	score
$L_{\texttt{cb}}$	5e-05	32	76.44	$L_{\texttt{fl}}$	5e-05	32	48.76
$L_{\texttt{ldam}}$	1e-05	32	76.13	$L_{\texttt{ce}}$	5e-05	32	48.49
$L_{\texttt{ldam}}$	1e-05	16	76.07	$L_{\texttt{fl}}$	5e-05	32	47.79
$L_{\texttt{fl}}$	5e-05	64	75.60	$L_{\texttt{fl}}$	5e-05	64	47.56
$L_{\texttt{ldam}}$	3e-05	16	75.59	$L_{\texttt{eq}}$	5e-05	32	47.10
$L_{\texttt{wce}}$	1e-05	32	75.57	$L_{\texttt{ce}}$	5e-05	64	47.07
$L_{\texttt{cb}}$	1e-05	32	75.55	$L_{\texttt{fl}}$	3e-05	32	47.07
$L_{\texttt{cb}}$	5e-05	64	75.37	$L_{\texttt{fl}}$	3e-05	16	46.51
$L_{\texttt{cb}}$	1e-05	32	75.23	$L_{\texttt{eq}}$	3e-05	32	46.50

Table 3: Best configurations on test set for each subtask. In all configurations, the models were trained on a concatenation of text and title for 50 epochs.

tions, for which we consider the best 9 configurations based on the validation scores (See table 5) for each of the four categories by using the majority voting technique. Using these top 9 configurations, we obtain predictions on the test set. We separately prepare ensembles for **Subtask1** and **Subtask2** by adding one by one configurations based on stopping criteria of validation scores. For **Subtask1**, we submitted the ensemble of top-9 configurations from Table 5. For *product-category*, we used $L_{\texttt{ldam}} + L_{\texttt{cb}} + L_{\texttt{wce}}$; for *hazard-category*, we used $L_{\texttt{cb}} + L_{\texttt{ldam}} + L_{\texttt{wce}} + L_{\texttt{ce}} + L_{\texttt{eq}}$. And, for **Subtask2**, we submitted the ensemble of top-7 configurations from Table 5. For *product*, we used $L_{\texttt{fl}} + L_{\texttt{eq}} + L_{\texttt{ce}}$ and for *hazard*, we used $L_{\texttt{ldam}} + L_{\texttt{eq}} + L_{\texttt{fl}} + L_{\texttt{ce}}$.

For **Subtask1**, in the pool of total 27 submissions for the test set, our final submission scored +5% more than the overall `Mean` submission and also more the `Median` submission. However, the best submission outperformed ours by approximately 4%. Overall, our submission was ranked 11th out of 27 and 9th among the 20 systems that used only `title+text` as input.

For **Subtask2**, our submission outperformed the `Mean` by +11% and the `Median` by approximately 1%. However, our submission fell short of the best system by 6%. Overall, our submission was ranked 6th in the pool of 26 submissions and 5th among the 18 systems that used only `title+text` as input.

During error analysis, we investigate our submission for the *product* task, which exhibits a unique disparity between the training set and the test set, as there are 82 out-of-distribution (OOD) classes present in the test set (see fig. 2). Upon further examination of the effectiveness of the system, we find that the model was unable to identify any of those classes. This is one of the clear limitations of loss function-based strategies for addressing the long-tailed imbalance problem.

	Subtask1	Subtask2
Best	**82.23**	**54.73**
Mean	$73.15_{11.5}$	$37.32_{16.7}$
Median	77.37	47.83
Ours	78.15	48.48
`title+text` Rank	9^{th} /20	5^{th} /18
`Overall` Rank	11^{th} /27	6^{th} /26

Table 4: Overview of the final leaderboard. `Overall` Rank corresponds to final leaderboard ranking provided by the Shared Task Organizers, whereas `title+text` Rank is overall ranking filtered by teams which use only `title` and `text` as inputs.

6 Conclusion

In this paper, we presented our submission to SemEval-2025 Task 9: Food Hazard Detection Challenge, where we focused on addressing the challenge of heavy class imbalance in the provided dataset. Our approach was designed to improve model performance despite the skewed class distribution. By exploring and implementing modifications to the loss function, we showed how techniques commonly used in visual recognition tasks to handle long-tail distributions can also be effectively applied to text classification.

Limitations

Loss functions such as Focal Loss, Equalization Loss, and Class-Balanced Loss address class imbalance but exhibit notable limitations in long-tail imbalanced settings, particularly in case of dealing with unseen test classes also referred to as out-of-domain (OOD) generalization (Zhang et al., 2023). First, these losses rely on static class weight adjustments, which fail to adapt when encountering domain shifts or evolving data distributions. Second, rare classes in the training set may be entirely absent in OOD settings, making prior class-based reweighting ineffective. Focal Loss, which emphasizes misclassified examples, may overfit to domain-specific hard samples, worsening OOD performance. Similarly, Equalization Loss, designed to suppress frequent class gradients, may lead to biased learning when class distributions change in

a new domain. Overall, while these loss functions improve in-domain class balance, they lack adaptability to unseen data, requiring complementary techniques such as contrastive learning, domain adaptation, and meta-learning for robust NLP classification in OOD scenarios.

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? Deep learning to the rescue! In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 34, pages 7383–7390.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32.

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.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277.

Ruofan Hu, Dongyu Zhang, Dandan Tao, Thomas Hartvigsen, Hao Feng, and Elke Rundensteiner. 2022. Tweet-fid: An annotated dataset for multiple foodborne illness detection tasks. *Preprint*, arXiv:2205.10726.

Bolei Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Liangliang Cao, and Stefanie Jegelka. 2020. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations (ICLR)*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Adyasha Maharana, Kunlin Cai, Joseph Hellerstein, Yulin Hswen, Michael Munsell, Valentina Staneva, Miki Verma, Cynthia Vint, Derry Wijaya, and Elaine O Nsoesie. 2019. Detecting reports of unsafe foods in consumer product reviews. *JAMIA Open*, 2(3):330–338.

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.

Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. 2020. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11662–11671.

Dandan Tao, Ruofan Hu, Dongyu Zhang, Jasmine Laber, Anne Lapsley, Timothy Kwan, Liam Rathke, Elke Rundensteiner, and Hao Feng. 2023. A novel foodborne illness detection and web application tool based on social media. *Foods*, 12(14).

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. 2017. Learning to model the tail. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. 2023. Deep long-tailed learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9):10795–10816.

Appendix

A Hyper-parameter search

In order to perform hyper-parameter search individually for each loss function, we considered different configurations. For learning rate (lr), we considered 1e-5, 3e-5 and 5e-5. For batch size (bs), we considered 16, 32 and 64.

Effect of Learning Rate. Overall, considering the evaluation criteria for scores for Subtask1 and Subtask2, the best learning rate was 3e-5.

Effect of Batch Size. Overall, based on the evaluation criteria for scores of Subtask1 and Subtask2, the best batch size was 32.

	product-category				**hazard-category**				**product**				**hazard**		
	lr	bs	seed		lr	bs	seed		lr	bs	seed		lr	bs	seed
$L_{\texttt{ldam}}$	3e-05	16	7897	$L_{\texttt{cb}}$	1e-05	16	45689	$L_{\texttt{fl}}$	3e-05	16	45689	$L_{\texttt{eq}}$	3e-05	64	45689
$L_{\texttt{cb}}$	1e-05	16	45689	$L_{\texttt{ldam}}$	1e-05	16	7897	$L_{\texttt{eq}}$	3e-05	16	7897	$L_{\texttt{ldam}}$	3e-05	16	45689
$L_{\texttt{ldam}}$	5e-05	64	7897	$L_{\texttt{wce}}$	3e-05	16	45689	$L_{\texttt{ce}}$	3e-05	16	7897	$L_{\texttt{ldam}}$	3e-05	64	45689
$L_{\texttt{ldam}}$	1e-05	16	7897	$L_{\texttt{fl}}$	3e-05	64	7897	$L_{\texttt{ce}}$	5e-05	16	78907	$L_{\texttt{ce}}$	3e-05	16	45689
$L_{\texttt{wce}}$	1e-05	32	78907	$L_{\texttt{cb}}$	1e-05	64	45689	$L_{\texttt{fl}}$	5e-05	32	7897	$L_{\texttt{fl}}$	3e-05	32	45689
$L_{\texttt{wce}}$	3e-05	32	45689	$L_{\texttt{ce}}$	3e-05	32	7897	$L_{\texttt{fl}}$	3e-05	16	7897	$L_{\texttt{eq}}$	5e-05	64	45689
$L_{\texttt{ldam}}$	3e-05	16	45689	$L_{\texttt{eq}}$	3e-05	32	45689	$L_{\texttt{eq}}$	5e-05	64	7897	$L_{\texttt{eq}}$	3e-05	32	7897
$L_{\texttt{wce}}$	1e-05	32	7897	$L_{\texttt{eq}}$	3e-05	16	45689	$L_{\texttt{ce}}$	5e-05	64	7897	$L_{\texttt{ldam}}$	5e-05	64	45689
$L_{\texttt{cb}}$	3e-05	16	7897	$L_{\texttt{ce}}$	1e-05	32	45689	$L_{\texttt{eq}}$	3e-05	16	45689	$L_{\texttt{ldam}}$	3e-05	32	45689

Table 5: Top-9 configurations for each task based on validation scores.