SemEval-2025 Task 9: The Food Hazard Detection Challenge

Korbinian Randl,[1] **John Pavlopoulos,**[1,2,3] **Aron Henriksson,**[1]
Tony Lindgren,[1] **Juli Bakagianni**[4]

[1] Stockholm University, Borgarfjordsgatan 12, 164 07 Kista, Sweden
{korbinian.randl,ioannis,aronhen,tony}@dsv.su.se
[2]Athens University of Economics and Business, Greece
[3]Archimedes, Athena Research Center, Greece
[4]Agroknow, Greece

Abstract

In this challenge, we explored text-based food hazard prediction with long tail distributed classes. The task was divided into two subtasks: **(1)** predicting whether a web text implies one of ten food-hazard categories and identifying the associated food category, and **(2)** providing a more fine-grained classification by assigning a specific label to both the hazard and the product. Our findings highlight that large language model-generated synthetic data can be highly effective for oversampling long-tail distributions. Furthermore, we find that fine-tuned encoder-only, encoder-decoder, and decoder-only systems achieve comparable maximum performance across both subtasks. During this challenge, we gradually released (under CC BY-NC-SA 4.0) a novel set of 6,644 manually labeled food-incident reports.

1 Introduction

The Food Hazard Detection Challenge at SemEval 2025 evaluated classification systems for titles of food-incident reports collected from the world wide web. Algorithms like these could, for example, be used to help automated crawlers find and extract food issues from publicly available sources like social media. Since such systems could have a high economic impact (specific food items may need to be recalled, leading to financial damage for the producers), transparency is extremely important. Human experts using data from these crawlers need to be well-informed about how the respective food issues are extracted.

Prior work has shown that a major challenge in food-hazard and food-product classification from text is the large number of possible classes, combined with a long-tail distribution (Randl et al., 2024b). To address this, we define two subtasks:

- **Subtask 1 (ST1)** focuses on training models for coarse-grained "category" prediction.
- **Subtask 2 (ST2)** is a more fine-grained "vector" prediction task.

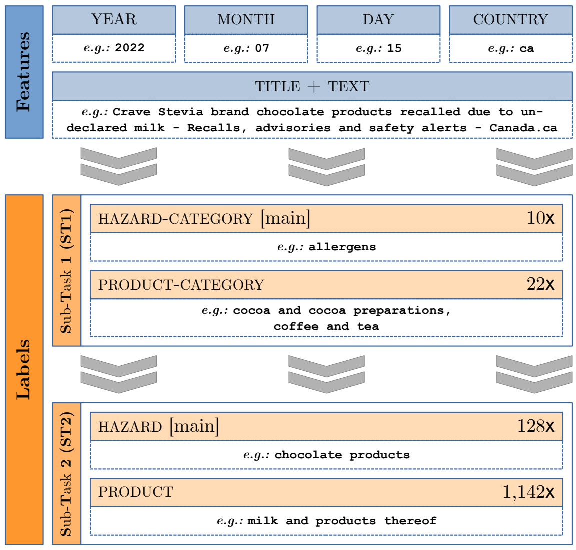

Figure 1: The columns in the blue boxes were available to the participants to serve as model input, while the orange boxes comprised the ground truth labels per subtask. The number on the right of each label indicated the number of unique values per label.

A prior SemEval challenge by Kirk et al. (2023) framed a similar setup as an initial step toward explainability. While this interpretation may be somewhat broad, we recognize that a "vector" prediction task is particularly valuable for automated information extraction, as it provides more specific information.

An overview of the SemEval-Task is shown in Figure 1. It includes **two sub-tasks**: **(ST1)** text classification for food hazard prediction, predicting the type of hazard (HAZARD-CATEGORY) and the type of product (PRODUCT-CATEGORY); **(ST2)** food hazard and product "vector" detection, predicting the exact hazard (HAZARD) and product (PRODUCT). The task was primarily concerned with detecting the hazard (more important than the product), hence a two-step scoring metric based

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 2523–2534
July 31 - August 1, 2025

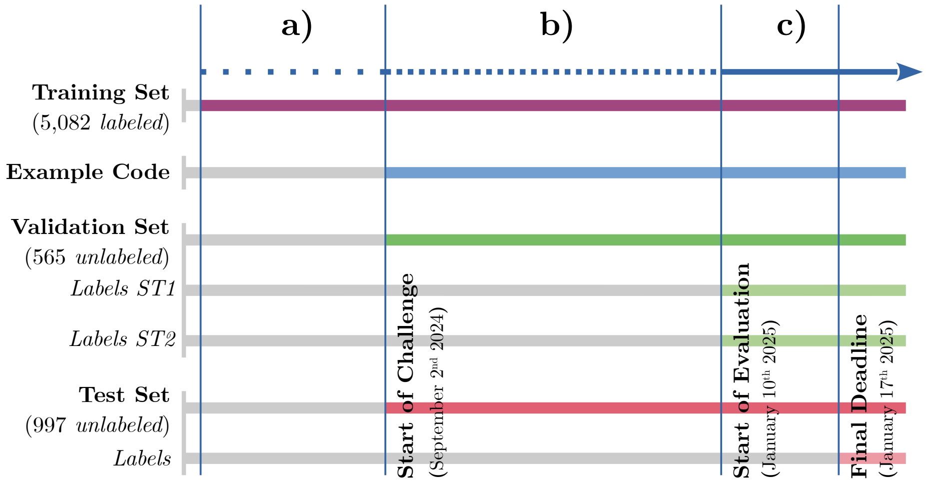

Figure 2: Timeline of the challenge: **(a) Trial Phase**: Training data was provided before the challenge commenced. **(b) Conception Phase**: Example code, along with unlabeled validation and test data, was released at the beginning of the challenge. During this phase, participants could submit separate trial entries for ST1 (category classification) and ST2 ("vector" classification) using the validation data. **(c) Evaluation Phase**: The validation data was made available, and final submissions for both tasks were accepted on the test data to determine the final ranking.

on the macro F_1 score was used, focusing on the respective hazard label per sub-task (see Section 4).

2 Task Organization

The detailed timeline of the project is illustrated in Figure 2. Participants were provided with training and validation data to develop, train, and evaluate their systems before the evaluation phase. The challenge was conducted on Codalab[1] (Pavao et al., 2023), adhering to the framework of previous competitions (Kirk et al., 2023).

The validation data was made available at the start of the challenge, enabling participants to submit to the leaderboards and compare their systems during the conception phase. However, these rankings did not influence the final results. The test set was released at the beginning of the challenge with labels concealed until its conclusion. During the evaluation phase, models could be trained on both the training and validation data but were evaluated exclusively on the test set to get the final ranking. After the evaluation phase, participants were required to submit a brief system description specifying the dataset features used, with this information made public alongside the final ranking.

Participants could submit up to five times per day and 100 times in total during the conception phase, whereas in the evaluation phase, each participant was limited to a single valid submission.

[1]https://codalab.lisn.upsaclay.fr

"Randsland brand Super Salad Kit recalled due to Listeria monocytogenes"	
`hazard`:	listeria monocytogenes
`hazard-category`:	biological
`product`:	salads
`product-category`:	fruits and vegetables
"Create Common Good Recalls Jambalaya Products Due To Misbranding and Undeclared Allergens"	
`hazard`:	milk and products thereof
`hazard-category`:	allergens
`product`:	meat preparations
`product-category`:	meat, egg and dairy products
"Nestlé Prepared Foods Recalls Lean Cuisine Baked Chicken Meal Products Due to Possible Foreign Matter Contamination"	
`hazard`:	plastic fragment
`hazard-category`:	foreign bodies
`product`:	cooked chicken
`product-category`:	prepared dishes and snacks

Table 1: Sample of texts along with their labels.

Additionally, participants were required to share their code (e.g., via GitHub) along with their system description papers.

3 Dataset

The dataset we used in the challenge is a subset of the data described in Randl et al. (2024b) and publicly accessible on zenodo (Randl et al., 2024a). It consists of 6,644 TITLEs (length in characters: *min=5, avg=88, max=277*), and full TEXTs (length in characters: *min=56, avg=2329, max=48318*) of

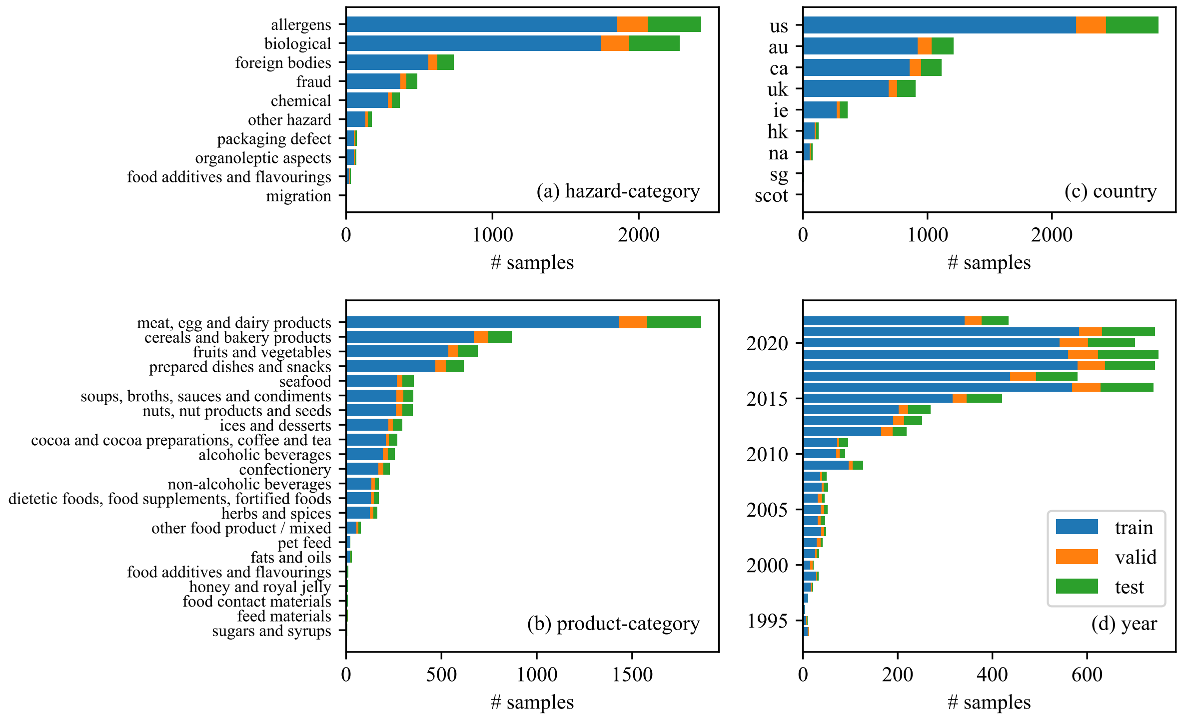

Figure 3: Overview over the data used in the challenge

English food recall announcements from the official websites of food agencies (e.g. the FDA's website). In addition, the dataset contains meta information such as date of download and country of issue. These texts were primarily gathered between 2012 and 2022 from domains based in the United States, Australia, Canada, and the United Kingdom (see Figure 3 (c) and (d)). The data was manually labeled with the reason for recall (HAZARD) and the recalled PRODUCT. Although neither TITLE nor TEXT individually is guaranteed to contain information about the product and hazard involved in a recall, their combination reliably provides the necessary details for classification. The distribution of this information between TITLE and TEXT varies across the dataset and largely depends on the issuing authority. Each pair of TITLE and TEXT has been assessed by two experts on food science or food technology from Agroknow[2]. Some sample TITLEs are shown in Table 1.

The data was stratified based on the more important hazard "vectors" (HAZARD) and divided into three subsets: 5,082 samples for training, 565 for validation, and 997 for evaluation. The training data, which was already published on zenodo (Randl et al., 2024a), also contains additional non-English texts that could be used by participants to train their classifiers. Nevertheless, our evaluation was only based on English texts. As the texts contain varying degrees of information on the HAZARD, we considered careful pre-processing of the data as part of the challenge. Upon completion of the task, the complete dataset was made available under the Creative Commons BY-NC-SA 4.0 license.

One sample of the dataset is shown in Figure 1. As described above, the data includes the features YEAR, MONTH, DAY, COUNTRY, TEXT and TITLE. Participants performed their text analysis primarily on the TITLE or TEXT fields, while additional features were available if needed. The task was to predict the labels PRODUCT-CATEGORY and HAZARD-CATEGORY, as well as the vectors PRODUCT and HAZARD. The dataset comprises 1,256 different PRODUCT values (e.g., "*ice cream*," "*chicken based products*," "*cakes*") sorted into 22 categories (e.g. "*meat, egg and dairy products*," "*cereals and bakery products*," "*fruits and vegetables*") with the help of ontologies. In addition, there are 261 distinct values for HAZARD (e.g., "*salmonella*," "*listeria monocytogenes*," "*milk and products thereof*") , which are grouped (again using ontologies) into the following 10 values of the la-

[2] https://agroknow.com

bel HAZARD-CATEGORY: "*allergens*," "*biological*," "*foreign bodies*," "*fraud*," "*chemical*," "*other hazard*," "*packaging defect*," "*organoleptic aspects*," "*food additives and flavourings*," "*migration*." The class distribution in the data is heavily imbalanced with the above examples being ranked from the most to the least common in Figure 3.

3.1 Baselines

In our challenge, we provided participants with three jupyter-notebooks for training and evaluating baseline models for both subtasks[3]:
(i) We provide a traditional pipeline consisting of a TF-IDF embedding in combination with a logistic regression classifier based on the `scikit-learn` Python module (Pedregosa et al., 2011).
(ii) A second baseline implementation fine-tunes an encoder-only transformer, specifically `bert-base-uncased` (Devlin et al., 2019), using the `transformers` Python module (Wolf et al., 2020) by huggingface.co.
(iii) Finally we provide a more sophisticated baseline based on the CICLe method (Randl et al., 2024b). It relies on prompting larger transformers such as GPT-4 without further fine-tuning (Brown et al., 2020) in combination with conformal prediction (Vovk et al., 2005). In our baseline we use the `crepes` Python module to implement conformal prediction (Boström, 2022).

Baseline performance of different classifiers on the whole dataset used in this challenge was also reported by Randl et al. (2024b). The results showed that the classification of hazards and products was a non-trivial task, and the classification of the "vector"-label, which we aimed to address in this challenge, was particularly challenging.

4 Evaluation

We computed the performance for ST1 and ST2 by calculating the macro F_1-score on the participants' predicted labels $\hat{\mathbf{y}}$ using the annotated labels $\mathbf{y}$ as ground truth. This measure is the unweighted mean of per-class-F_1-scores over the n classes. Both $\hat{\mathbf{y}}$ and $\mathbf{y}$ are vectors of m samples:

$$\mathrm{F}_1(\mathbf{y}, \hat{\mathbf{y}}) = \frac{2}{n} \sum_{i=0}^{n} \frac{\mathrm{RCL}_i(\mathbf{y}, \hat{\mathbf{y}}) \cdot \mathrm{PRC}_i(\mathbf{y}, \hat{\mathbf{y}})}{\mathrm{RCL}_i(\mathbf{y}, \hat{\mathbf{y}}) + \mathrm{PRC}_i(\mathbf{y}, \hat{\mathbf{y}})} \quad (1)$$

where RCL_c is the recall and PRC_c is the precision for a specific class c. In order to combine the predictions for the HAZARD and PRODUCT labels into one score, we took the average of the scores:

$$\mathrm{S}(Y, \hat{Y}) = \frac{\mathrm{F}_1(\mathbf{y}^h, \hat{\mathbf{y}}^h) + \mathrm{F}_1(\mathbf{y}^{p|h}, \hat{\mathbf{y}}^{p|h})}{2} \quad (2)$$

Here $Y = [\mathbf{y}^h, \mathbf{y}^p]$ is the $2 \times m$ matrix with the HAZARD label $\mathbf{y}_h$ and the PRODUCT label $\mathbf{y}_p$ as column vectors. The vector $\mathbf{y}^{p|h}$ is defined as the entries of $\mathbf{y}^p$ where $\mathbf{y}^h$ is correctly predicted:

$$\mathbf{y}^{p|h} = \{\mathbf{y}_j^p \mid \hat{\mathbf{y}}_j^h = \mathbf{y}_j^h\},\ j \in \{1, 2, ..., m\} \quad (3)$$

The scalar $\mathbf{y}_j^*$ is the j-th element of $\mathbf{y}^*$. $\hat{Y}$ and $\hat{\mathbf{y}}^{p|h}$ are defined accordingly. With this measure we based our rankings predominantly on the predictions for the HAZARD classes. Intuitively, this means that a submission with both $\mathbf{y}^h$ and $\mathbf{y}^p$ completely right would have scored 1.0, a submission with $\mathbf{y}^h$ completely right and $\mathbf{y}^p$ completely wrong would have scored 0.5, and any submission with $\mathbf{y}^h$ completely wrong would have scored 0.0 independently of the value of $\mathbf{y}^p$.

5 Participant Systems and Results

In total, our task attracted approximately 260 participants and received 99 valid submissions during the evaluation phase. Among these, 27 system description papers were submitted for peer-review. These 27 systems form the basis of our analysis and the official ranking, as they are accompanied by detailed system descriptions, enabling a thorough evaluation. The full, unofficial ranking – including all submissions to codalab – is available on the task's website.[4]

5.1 Popular Methods

Figure 4 illustrates the frequency distribution of system attributes. Each subplot corresponds to a distinct attribute, highlighting key trends among the systems. We observe that the majority (16 systems) of systems uses both TITLE and TEXT features, while three systems incorporated all available dataset features. Furthermore, the majority (21 systems) treated the tasks separately, with only five systems leveraging a combined approach to exploit the correlation between the tasks. In terms of model choice, most systems (19) relied on encoder-only transformer models, while two used traditional machine learning models. Among the systems that

[3] `https://food-hazard-detection-semeval-2025.github.io/code/`

[4] `https://food-hazard-detection-semeval-2025.github.io/`

used transformer-based models, open-source models (24 systems) were preferred. Furthermore, the majority (14 models) opted for a single model for classification rather than an ensemble strategy (11 systems). Finally, regarding the data sources, 12 systems incorporated synthetic data, for example oversampling with LLM-generated texts, to address the tasks.

5.2 Leaderboard Results

ST1 Table 2 presents the results and the ranking of the systems that submitted system a description paper in ST1. The scores lie between 0.1426 and 0.8223, with the largest gap in performance observed between the first and second-ranked systems among the top three. Systems ranked between fifth and 16th exhibit relatively similar scores, while a distinct widening of the gap is evident in the lower ranks. Furthermore, the top two systems used richer feature sets compared to the lower-ranked systems, indicating that the richer feature sets may have contributed to their scores, while most systems relied on both textual features, i.e., TITLE and TEXT, rather than focusing on one of them.

RANK	TEAM NAME	SCORE	FEATURES
Baselines:	TFIDF + LR	0.498	TITLE
	BERT	0.667	TITLE
1	Anastasia	0.8223	YEAR, MONTH, DAY, COUNTRY, TITLE, TEXT
2	MyMy	0.8112	YEAR, MONTH, DAY, COUNTRY, TITLE, TEXT
3	SRCB	0.8039	TITLE, TEXT
4	PATeam	0.8017	TITLE, TEXT
5	HU	0.7882	TITLE, TEXT
6	BitsAndBites	0.7873	TITLE, TEXT
7	CSECU-Learners	0.7863	TITLE, TEXT
8	ABCD	0.7860	TITLE, TEXT
9	MINDS	0.7857	TITLE, TEXT
10	Zuifeng	0.7835	TITLE
11	Fossils	0.7815	TITLE, TEXT
12	PuerAI	0.7729	TITLE
13	Ustnlp16	0.7654	TITLE, TEXT
14	FuocChu_VIP123	0.7646	TEXT
15	BrightCookies	0.7610	TEXT
16	farrel_dr	0.7587	TITLE, TEXT
17	OPI-DRO-HEL	0.7381	TITLE, TEXT
18	madhans476	0.7362	TITLE, TEXT
19	Anaselka	0.6858	TITLE, TEXT
20	Somi	0.6614	TITLE, TEXT COUNTRY, TITLE, TEXT
21	TechSSN3	0.6442	TEXT
22	UniBuc	0.6355	TITLE, TEXT
23	CICL	0.6079	TEXT
24	VerbaNexAI	0.5165	TITLE
25	JU-NLP	0.4566	TITLE, TEXT
26	Habib University	0.4482	TITLE, TEXT
27	Howard University-AI4PC	0.1426	TEXT

Table 2: ST1 ranking for systems of teams that submitted a system description paper. Gray entries are outperformed by the best baseline.

ST2 Table 3 presents the results and the rankings of the systems for ST2 that submitted a system description paper. The results show significantly lower performance compared to ST1, with the highest score of SRCB (0.5473) being considerably lower than the top score in ST1 (0.8223), which indicates that ST2 is a more challenging task. A sharp drop in scores is observed after the top three teams and again after the 12th team (BitsAndBites), with the lowest-ranked system (Anaselka) receiving 0.0049 score. Notably, the top three teams in both subtasks – except for the Anastasia team, which focused only on ST1 – performed well in both, consistently ranking among the top teams in each subtask. Among the top 15 systems, while a few systems, such as Anastasia and BitsAndBites, performed better on ST1, a larger number of systems, including MINDS, Fossils, PuerAI, and BrightCookies, achieved significantly higher rankings in ST2.

RANK	TEAM NAME	SCORE	FEATURES
Baselines:	TFIDF + LR	0.183	TITLE
	BERT	0.165	TITLE
1	SRCB	0.5473	TITLE, TEXT
2	MyMy	0.5278	YEAR, MONTH, DAY, COUNTRY, TITLE, TEXT
3	PATeam	0.5266	TITLE, TEXT
4	HU	0.5099	TITLE, TEXT
5	MINDS	0.4862	TITLE, TEXT
6	Fossils	0.4848	TITLE, TEXT
7	CSECU-Learners	0.4797	TITLE, TEXT
8	PuerAI	0.4783	TITLE
9	Zuifeng	0.4712	TITLE
10	ABCD	0.4576	TITLE, TEXT
11	BrightCookies	0.4529	TEXT
12	Ustnlp16	0.4512	TITLE, TEXT
13	BitsAndBites	0.4456	TITLE, TEXT
14	UniBuc	0.3453	TITLE, TEXT
15	OPI-DRO-HEL	0.3295	TITLE, TEXT
16	VerbaNexAI	0.3223	TITLE
17	CICL	0.3169	TEXT
18	Somi	0.3048	TITLE, TEXT COUNTRY, TITLE, TEXT
19	TechSSN3	0.2712	TEXT
20	Howard University-AI4PC	0.1380	TEXT
21	Anastasia	0.1281	YEAR, MONTH, DAY, COUNTRY, TITLE, TEXT
22	farrel_dr	0.1249	TITLE, TEXT
23	madhans476	0.0486	TITLE, TEXT
24	Habib University	0.0315	TITLE, TEXT
25	JU-NLP	0.0126	TITLE, TEXT
26	Anaselka	0.0049	TITLE, TEXT

Table 3: ST2 ranking for systems of teams that submitted a system description paper. Gray entries are outperformed by the best baseline.

5.3 Best Systems

In this section, we outline the key methods employed by the top three systems for each subtask.

Figure 4: Frequency distribution of system attributes. Each subplot represents a distinct attribute, illustrating the choices made by the participating systems in terms of features, task treatment, model types, ensemble strategies, model availability, and data usage.

Since the teams "MyMy" and "SCRB" rank among the top three in both evaluations (see Tables 2 and 3), we analyze a total of four systems.

SRCB (**ST1:** 3^{rd}, **ST2:** 1^{st}) The first place in ST2 comes from Ricoh Software Research Center. Zhang et al. (2025) concatenated the TITLE and the TEXT in lower case, and followed a two-step approach. In the first step, they used BERT to reduce the label space and include only the most probable ones. In a second step, then, all the possible labels (possibly along with examples) were fed to a large language model (LLM) to predict the correct one. This approach follows the paradigm of Randl et al. (2024b), who suggested reducing the possible labels by quantifying the uncertainty with conformal prediction (Vovk et al., 2005). Infrequent categories were furthermore augmented with an LLM, while approx. 10% of the data was truncated.

Anastasia (**ST1:** 1^{st}, **ST2:** 21^{th}) The best system in ST1 is by Le et al. (2025) from VNUHCM – University of Information Technology and focuses on ST1, while neglecting ST2. After a simple text normalization step, they chunk the texts into snippets of consecutive sentences that fit the context windows of their applied models. Following this, they fine-tune two encoder-only transformers, specifically `DeBERTa-v3-large` and `RoBERTa-large`, using focal loss (Lin et al., 2017) with class weights. For training, they compare two setups: **(i)** They try **multi-task** fine-tuning of `DeBERTa-v3-large` to get a combined model for both HAZARD and PRODUCT prediction using oversampling for underrepresented classes and undersampling for overrepresented classes. **(ii)** Additionally, they try **single-task** fine-tuning of both `DeBERTa-v3-large` and `RoBERTa-large`, this time addressing the class-imbalance by creating synthetic samples by prompting `gemini-2.0-flash-exp` to paraphrase texts in underrepresented classes. They report that multi-task training leads to slightly worse performance on ST1 compared to single-task. This may be owed to the different resampling approaches, though. Finally, they combine all of their trained models (single- and multi-task) in one ensemble, using soft voting with a weighted sum. The weights were based on grid search on the validation set.

MyMy (**ST1:** 2^{nd}, **ST2:** 2^{nd}) Phan and Chiang (2025) from the Department of Computer Science and Information Engineering, National Cheng Kung University employ a retrieval-augmented generation (RAG) approach to address both subtasks separately by intergrating domain-specific external knowledge. It first retrieves relevant documents for each data sample from PubMed,[5] following the RAG paradigm: it uses GPT- 3.5 Turbo3,[6] Gemini Flash 2.0 (Team et al., 2023), Llama 3.1 8B (Touvron et al., 2023), and Mistral 8x7B (Jiang et al., 2023) LLMs to simplify the original data sample; it then retrieves documents from the PubMed API, encodes them into embeddings using `nomic-embed-text-v1` (Nussbaum et al., 2024) and stores them in a Chroma embedding database; cosine similarity scores are then computed to retrieve the top-K most relevant documents. These

[5] `https://pubmed.ncbi.nlm.nih.gov/`
[6] `https://platform.openai.com/docs/models/gpt-3-5-turbo`

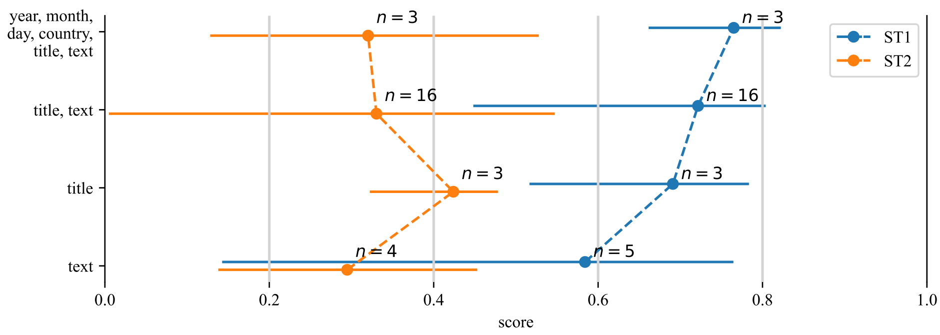

Figure 5: Average score achieved and number of submissions per combination of input features used (– ST1, – ST2). The horizontal bars show minimum and maximum score and the number of samples is annotated as n.

documents are then combined with the original input and paraphrased using the same LLMs to generate augmented data. A validation step incorporating the same LLMs is used to filter the generated samples based on relevance, ensuring data quality. The enriched dataset is then used to fine-tune classification models (Gemini Flash 2.0 (Team et al., 2023), PubMedBERT (Gu et al., 2021), and ModernBERT (Warner et al., 2024)). Finally, predictions are obtained through a weighted soft voting strategy, where class probabilities from multiple models are combined using weighted sums to determine the final label.

PATeam (**ST1:** 4^{th}, **ST2:** 3^{rd}) Wan et al. (2025) begin with data cleaning using regular expressions, followed by text augmentation, where LLM-generated summaries are concatenated with the TEXT feature. To address data imbalance, SMOTE (Chawla et al., 2002) is applied to underrepresented categories (fewer than five samples, a threshold determined through tuning) to ensure a minimum of five samples per class. The system employs a bagging approach with bootstrapping to generate five subsets of the training data, fine-tuning five `microsoft/phi-4` models[7] using low-rank adaptation (LoRA) (Hu et al., 2021) to reduce trainable parameters. Predictions from all five models are integrated via an ensemble voting mechanism. The system employs the multi-dimensional type-slot label interaction network (MTLN) (Wan et al., 2023) to capture the correlation between the two subtasks. It first classifies ST1 and then utilizes these predictions to inform the classification of ST2. An ablation study confirmed that this multi-task approach outperforms treating the tasks independently.

[7] https://huggingface.co/microsoft/phi-4

5.4 What Worked Well

A prevalent strategy among these systems is the use of generative LLMs for synthetic data creation to mitigate class imbalance. Specifically, three approaches stand out: (i) **paraphrasing** (Le et al., 2025), (ii) **summarizing** and appending generated text to the original (Wan et al., 2025), and (iii) **generating** new samples by combining information from two instances of the same class (Zhang et al., 2025). Additionally, Le et al. (2025) and Phan and Chiang (2025) incorporate class-weighted loss functions to increase the impact of underrepresented classes during training.

Another common technique among top-ranking systems is the use of ensemble methods. Le et al. (2025) employ a soft voting approach, optimizing weights of different models during grid search on a validation set, while Phan and Chiang (2025) adopt a max voting strategy, selecting the prediction from the most confident model. Wan et al. (2025) fine-tune five classifiers using bootstrapped subsets of their preprocessed training data.

In contrast to these shared strategies, there is no clear consensus on the use of multi-task learning (joint modeling of both subtasks) versus single-task learning (treating subtasks separately). Three out of four systems opt for a single-task approach, but Wan et al. (2025) experiment with both strategies in a prompting-based classification setup. Their results suggest that multi-turn prompts, where both subtasks are addressed within a single interaction, outperform single-turn prompts, which handle the subtasks separately.

As discussed in Section 5.2, richer feature sets tend to support stronger models across both subtasks. This observation is further illustrated in Figure 5, where systems leveraging multiple input features consistently outperform those using only a single feature (according to maximum achieved score). Notably, models utilizing only TITLEs tend to achieve better results than those using only TEXTs. A plausible explanation is that TITLEs often contain more concise and targeted information compared to the broader and potentially noisier content in TEXTs. Interestingly, the three approaches that utilize all available features achieve better results in ST1, but underperform slightly in ST2, suggesting that their design was primarily optimized for the former.

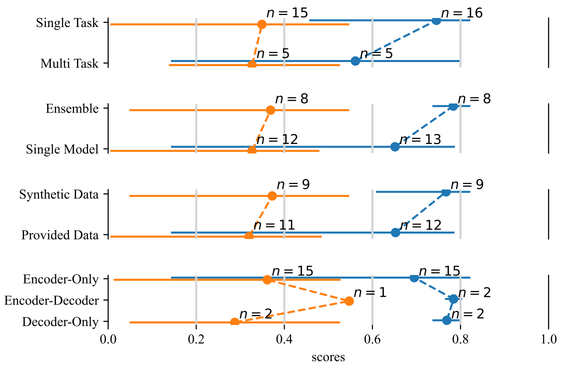

Figure 6: Average score achieved and number of submissions per combination per design choice (– ST1, – ST2). The horizontal bars show minimum and maximum.

Figure 6 presents a detailed comparison of design choices based on evaluation scores. Interestingly, treating the subtasks separately leads to better performance than multi-task approaches that use a shared model for ST1 and ST2. Additionally, leveraging an ensemble of multiple models proves more effective than relying on independent models.

As discussed in Section 3, one major challenge participants faced was the extreme class imbalance in the dataset. It is therefore unsurprising that oversampling underrepresented classes with artificially generated data significantly improved performance compared to using only the provided training set. As noted in Section 5.3, this artificial data was typically generated by prompting LLMs. Finally, an interesting finding is that no transformer architecture – whether encoder-only (e.g. BERT), encoder-decoder (e.g. BART), or decoder-only (e.g. Llama) – consistently outperforms the others. Across all three architectures, the highest achieved scores remain approximately equal within each subtask.

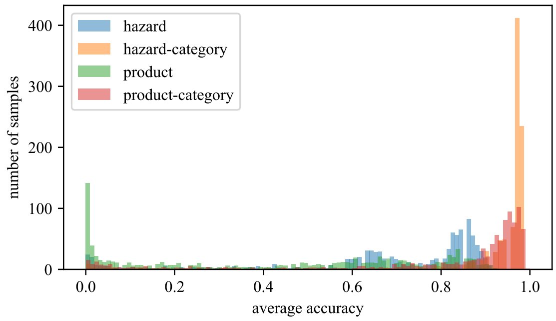

Figure 7: Histogram of the frequency (vertically) across the fraction of systems correctly predicting a specific sample (horizontally).

6 Discussion

6.1 Task Difficulty Estimation

We show an instance-based difficulty analysis in Figure 7. The figure shows that across categories/vectors most samples are more likely to be predicted correctly than not. Nevertheless, we also see a spike at zero accuracy, which is most prevalent for the vector PRODUCT, but seen for all categories/vectors. This indicates that several samples were never correctly classified, indicating that they are extremely difficult or even missing information. To make it easier to identify such instances in our data, we include an instance difficulty score, ranging linearly from 0 (*instance was classified correctly by all submissions*) to 1 (*instance was never correctly classified*), for all instances in the train and test set in our dataset on zenodo.

6.2 Error Analysis

Figure 8 shows the pairwise error rate between the submissions per category. The error is considerably higher in ST2 for the two plots on the right compared to the two of ST1. This is partly due to the fewer number of possible labels in the latter and the higher likelihood of mistakes on the former.

A more detailed analysis is shown in the confusion matrices in Appendix A. For HAZARD-CATEGORY, we see that precision and recall are relatively high except for the classes "*migration*" and "*food additives and flavourings*" (see Figure 9). While samples of the class "*migration*" are predicted to the very similar class "*chemical*" in 90% of the cases, predictions for "*food additives and flavourings*" are divided between the true class (49%), "*other hazard*" (28%), "*fraud*" (13%), and "*allergens*" (13%).

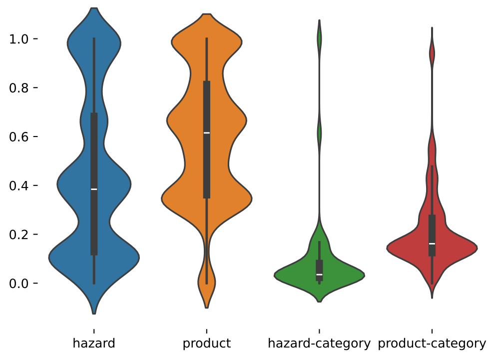

Figure 8: Pairwise error rate (vertically) of submissions (horizontally)

We see a similar picture for PRODUCT-CATEGORY in Figure 10. Most classes show good performance, while "*food additives and flavourings*," "*honey and royal jelly*," and "*other food product / mixed*" show high misclassification rates. "*Food additives and flavourings*" is most commonly confused with "*meat, egg and dairy products*" (22%) and "*cereals and bakery products*" (18%). "*Honey and royal jelly*" is confused with the most supported class "*meat, egg and dairy products*" in 40% of the cases. As an overarching class for leftover samples, "*other food product / mixed*" is misclassified to multiple other classes, most prominently "*soups, broths, sauces and condiments*" (18%), and "*fruits and vegetables*" (18%). All of these commonly mislabeled classes are highly underrepresented in the dataset and/or easy to confuse with other, higher-supported classes in the data.

7 Conclusion

In conclusion, our task demonstrates that LLM-generated synthetic data can be highly effective for oversampling in long-tail distributions. A second, albeit expected, finding is that ensemble strategies significantly enhance classification performance. Additionally, while combined approaches for vector and category classification can be beneficial in prompting scenarios, they do not generally lead to performance improvements. More notably, we do not observe a clear winner among transformer architectures: fine-tuned encoder-only, encoder-decoder, and decoder-only models achieve comparable maximum performance across both subtasks.

Future research on our dataset should prioritize the more challenging vector classification task. Our analysis indicates that classification errors often stem from low class support and that food recall texts contain ambiguous instances, with semantically similar classes contributing to misclassification. We argue that debugging classifiers using explainability techniques may help improve performance.

Despite its potential to assist human validation and enable meta-learning approaches, such as clustering or pre-sorting examples, explainability in text-based food risk classification remains underexplored. However, explanations can vary significantly depending on the model and task. Existing literature addresses both model-specific (Assael et al., 2022; Pavlopoulos et al., 2022) and model-agnostic (Ribeiro et al., 2016) explainability approaches, which should be further investigated in this domain.

Limitations

(i) A limitation of our evaluation process is that, while we enforced a one-submission-per-user policy during the evaluation phase, some participants have circumvented this by registering multiple accounts. We chose not to remove suspicious accounts, as identifying all of them would have been impractical and likely only encouraged more covert attempts to bypass the restriction.
(ii) We chose to release the unlabeled test set at the beginning of the challenge, as this was easier to set up with codalab. While this ensured transparency throughout the challenge, participants strongly determined to win could peak (e.g., manually annotating the test data).
(iii) We found 42 duplicate entries in our dataset after the start of the challenge. These were introduced due to an error in one of our preprocessing scripts and resulted in six entries that are present in both the training and validation set as well as seven entries that are present in both the training and test set. As this concerns less than 1% of the data, we argue that it is not severely impacting our results.

Ethical Statement

All texts are collected from official and publicly available sources, hence no privacy-related issues are present. All annotations have been provided by Agroknow experts. System application is intended to complement and not substitute the human expert in preventing illness or harm from food sources.

Acknowledgments

We thank Giannis Stoitsis and Agroknow for collecting and providing the data for this task.

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

This research has also been partially funded by the European Union's Horizon Europe research and innovation program EFRA (Grant Agreement Number 101093026). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or European Commission-EU. Neither the European Union nor the granting authority can be held responsible for them.

References

Y. Assael, B. Shillingford, M. Bordbar, N. de Freitas, T. Sommerschield, J. Pavlopoulos, M. Chatzipanagiotou, I. Androutsopoulos, and J. Prag. 2022. Restoring and attributing ancient texts using deep neural networks. *Nature*, 603(7900):280–283 – 283.

Henrik Boström. 2022. crepes: a python package for generating conformal regressors and predictive systems. In *Proceedings of the Eleventh Symposium on Conformal and Probabilistic Prediction and Applications*, volume 179 of *Proceedings of Machine Learning Research*. PMLR.

T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. 2020. Language models are few-shot learners.

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.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 4171–4186 – 4186.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Hannah Kirk, Wenjie Yin, Bertie Vidgen, and Paul Röttger. 2023. SemEval-2023 task 10: Explainable detection of online sexism. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2193–2210, Toronto, Canada. Association for Computational Linguistics.

Tung Thanh Le, Tri Minh Ngo, and Trung Hieu Dang. 2025. Anastasia at semeval-2025 task 9: Subtask 1, ensemble learning with data augmentation and focal loss for food risk classification. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 141–146, Vienna, Austria. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:318–327.

Zach Nussbaum, John X Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Dinh-Tuan Tran, Xavier Baro, Hugo Jair Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2023. Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research*, 24(198):1–6.

J. Pavlopoulos, A. Xenos, I. Androutsopoulos, L. Laugier, and J. Sorensen. 2022. From the detection of toxic spans in online discussions to the analysis of toxic-to-civil transfer. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 3721–3734 – 3734.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ben Phan and Jung-Hsien Chiang. 2025. Mymy at semeval-2025 task 9: A robust knowledge-augmented data approach for reliable food hazard detection. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 815–825, Vienna, Austria. Association for Computational Linguistics.

Korbinian Randl, Manos Karvounis, George Marinos, John Pavlopoulos, Tony Lindgren, and Aron Henriksson. 2024a. Food recall incidents.

Korbinian Randl, John Pavlopoulos, Aron Henriksson, and Tony Lindgren. 2024b. 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.

M.T. Ribeiro, S. Singh, and C. Guestrin. 2016. 'why should i trust you?' : Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135 – 1144.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. 2005. *Algorithmic Learning in a Random World*. Springer International Publishing.

Xue Wan, Fengping Su, Ling Sun, Yuyang Lin, and Pengfei Chen. 2025. Pateam at semeval-2025 task 9: Llm-augmented fusion for ai-driven food safety hazard detection. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 1902–1908, Vienna, Austria. Association for Computational Linguistics.

Xue Wan, Wensheng Zhang, Mengxing Huang, Siling Feng, and Yuanyuan Wu. 2023. A unified approach to nested and non-nested slots for spoken language understanding. *Electronics*, 12(7):1748.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Yuming Zhang, Hongyu Li, Yongwei Zhang, Shanshan Jiang, and Bin Dong. 2025. Srcb at semeval-2025 task 9: Llm finetuning approach based on external attention mechanism in the food hazard detection. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 999–1006, Vienna, Austria. Association for Computational Linguistics.

A Confusion Matrices

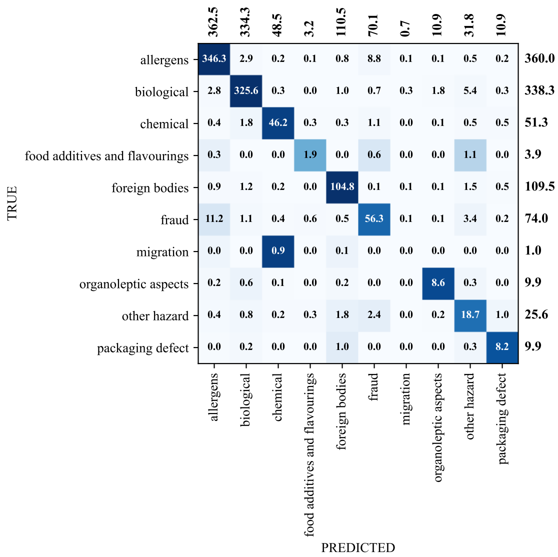

Figure 9: Confusion Matrix for HAZARD-CATEGORY. Numbers signify average number of occurrences per submission during the evaluation phase. Colors are normalized by row.

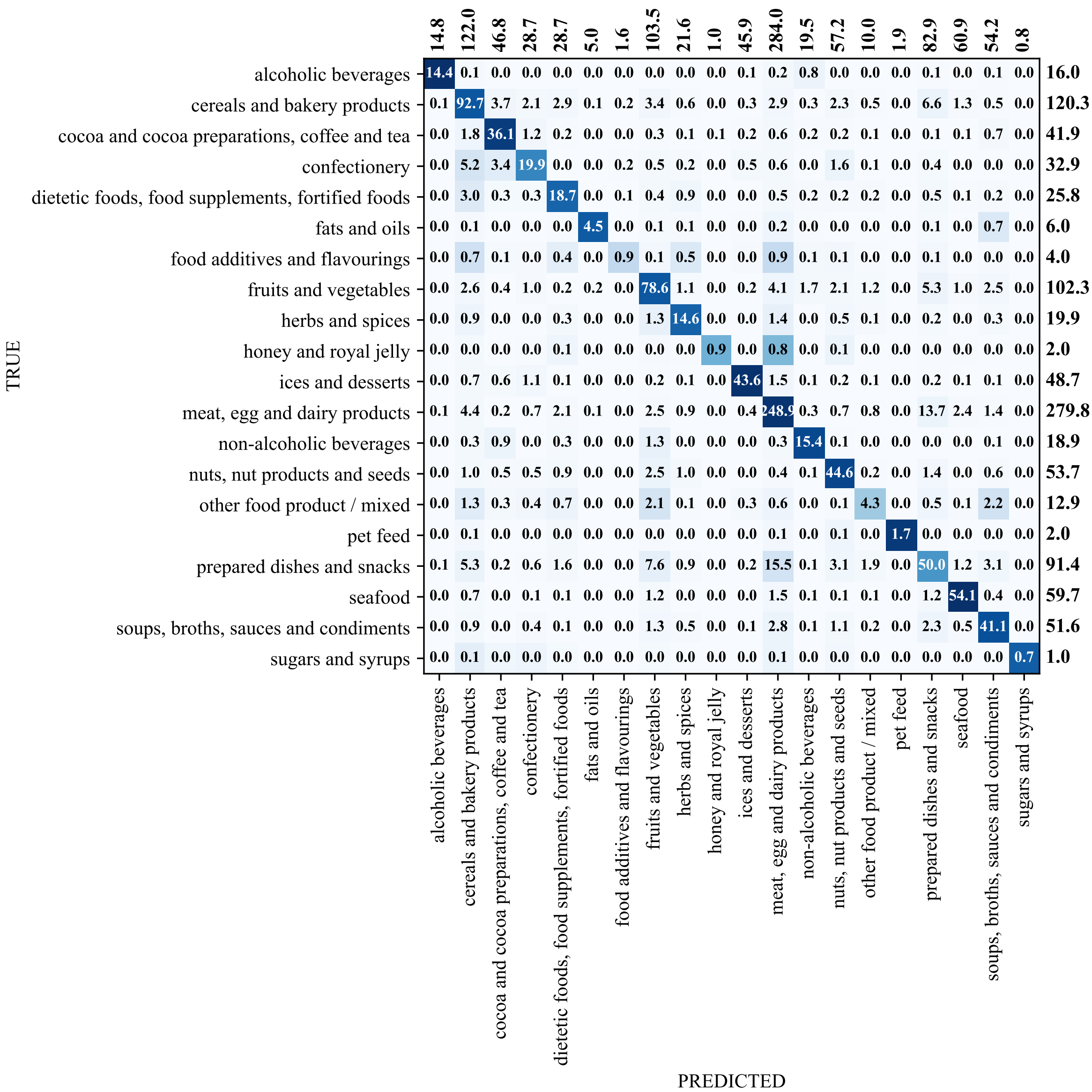

Figure 10: Confusion Matrix for PRODUCT-CATEGORY. Numbers signify average number of occurrence per submission during the evaluation phase. Colors are normalized by row.