SemEval 2025 Task 9: The Food Hazard Detection Challenge

Term Paper Submission for the Course
IT359 - Pattern Recognition
Odd 2024-25

by

Chinta Tejdeep Reddy(211AI013)
Praveen K (211AI028)
Savla Jay Paresh (211AI031)

under the guidance of

Dr. Shrutilipi Bhattacharjee

DEPARTMENT OF INFORMATION TECHNOLOGY
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE - 575025

August 2024

ABSTRACT

The Food Hazard Detection initiative aims at developing intelligent systems that can identify possible hazards in food products. And this should enhance public health and safety. With a growingly complex food industry the detection of harmful ingredients, contaminants, or unsafe conditions in foods becomes more important than ever. The validation of food hazards requires machine learning, data-driven analytics with classifications and natural language processing. The overall idea is to build a model that identifies food hazards and also predicts the hazard type and product. Ultimately, it aims to transform food safety policies, incorporating some of the current state-of-the-art technology in hazard identification, ensuring that there is a safe food supply without food-related health issues.

Keywords: Food Hazard Detection, Public health and safety, Machine learning, Natural language processing (NLP), Predictive modeling

Contents

List of Figures

List of Tables

1 Introduction

The global food industry has grown more complex due to longer supply lines and more variety. These are challenges that raise the risk of falling into foodborne illness, thus threatening public health and safety. Food hazards include natural agents, such as bacteria, viruses, and parasites; chemical pollutants, including pesticides, toxins, and allergens; and physical substances such as glass and metals. These hazards can lead to foodborne illness, allergies, chronic health issues, or even death if not identified and mitigated at the early stages of food production. Hazards in food follow-through is crucial to protecting consumers, ensuring food quality, and bringing trust to the food supply chain. Traditional methods of food hazard identification, such as manual inspection and laboratory testing, are usually insufficient for the scope and pace of food production and, more importantly, these methods are rather reactive than proactive, and they recognize hazards only after early contamination.

Foodborne disease is an important public health issue worldwide. According to the World Health Organization, an estimated 1 in 10 people suffer from foodborne illness each year, with some 600 million illnesses and 420,000 deaths, many of them young children under the age of five years. Addressing such challenges, we seek to apply machine learning and advanced data analytics leveraging big data and predictive algorithms, exactly the types of possibilities we have seen in real time. This paper looks to revamp food safety by helping the development of intelligent systems that identify food hazards and identify their nature and consequences.

2 Literature Survey

2.1 Motivation

The task of Food Hazard Detection will rank explainable classification systems by analyzing titles of food incident reports collected using open-source monitoring. Such algorithms will strongly increase the capability of automatic crawlers for the identification and the extraction of problems concerning food issues from social network platforms. Due to huge economic interests involved and problems that have occurred in the society, there is an urgency for the transparency of such systems. This underlines that the importance of our literature survey is as in inasmuch as it has scope to explore and review all the existing methods and their related developments within this field.

Dong Zhe et al, the paper: ”A Food Safety Text Filtering Method Based on Text Classification Techniques” [1]. Reports The problem of using advanced classification methods to extract food safety-related information from raw data. The study presents a new method combining BERT pretraining with the TF-IDF criteria and word vectors to enhance the performance of text classification. The procedure involves collecting and then preprocessing the data with a web crawler, optimization of the BERT model for a particular corpus, and then comes finally combines TF-IDF using embedded words to produce document vectors, which then trains the SVM classifier on those vectors to classify the food safety labels. Experiments results reveal that the proposed methods yields the better performance against traditional methods, and then the SVM gained greater accuracy (97.1 %) and accuracy (92.1 %) as compared to naive Bayes, decision tree, KNN, and random forest classifiers. The result is that TF-IDF with interpolation and advantages of advanced models like BERT has greatly improved the classifying of the information on outcomes in food safety issues.

Wang et al.

citeref2 talk about social media or Yelp reviews use for the prediction of foodborne diseases outbreaks in restaurants as an efficient way of restaurant inspections. Their work, ”Predictive Analytics Using Text Classification for Restaurant Inspections,” uses foodborne diseases as among the major public health concerns; millions experience it each year in the United States, therefore, constant follow-up inspections by departments of health require. They have published a predictive analytics framework with text mining and machine learning techniques through which they find the foodborne illness indicators using the Yelp reviews. Their methodology involves statistically and linguistically interpreting the features of reviews and categorizing them based on supervised machine learning models like Naïve Bayes, Support Vector Machines, Random Forests, and Recurrent Neural Networks. This ranged from good performances with Naïve Bayes and support vector machines with specific kernels to the most promising results achieved by the Recurrent Neural Network in accuracy and F-scores. Thus, a case of noisy and imbalanced data poses a challenge, and the integration of user reviews may create effective predictions of possible foodborne diseases or improvements in public health responses. It concludes by making recommendations on the integration of real-time data and semantic analysis to improve the prediction models.

The paper "CICLe: Conformal In-Context Learning for Largescale Multi-Class Food Risk Classification" [3], addresses the pressing challenge of contamination and adulteration of food in public circulation. Using a dataset of 7,546 short texts from 24 domains describing food recall announcements, this study performs classification both at a coarse and fine level of granularity into 261 hazard classes and 1,256 product classes. In general, for the few-shot classes, state-of-the-art models considered, all with RoBERTa (125M parameters) and XLM-R (270M parameters), always performed poorer than the traditional machine learning methods using Logistic Regression with TF-IDF representation in classes where support is low. All classes for which a few-shot prompt is needed to improve the detection are included in those methods whose F1-score was below 0.5 in the Transformer model. These challenges therefore motivate further, more in-depth studies that establish better classification results. A novel framework is proposed herein, **CICLe**, combining Conformal Prediction with few-shot prompting over GPT-3.5. CICLe The classification accuracy increases with much lower computation resources and energy consumption on which it runs. Narrowed to relevant classes and context length, it achieves traditional model accuracy using as much as 40% more energy efficiency. As the following research demonstrates, the resource-frugal integration of ML and LLMs developed by CICLe increases performance not only for classes with low support but also for record-breaking large-scale multi-class tasks in food risk categorization. General, this work gives a strong overview on how to improve the quality and efficiency of classification both in terms of accuracy for NLP tasks related to food safety. It has proposed a new dataset and its performance benchmarking accompanied by a novel methodological description on resource-efficient machine learning-large language model combination.

A paper would be "Food Safety News Events Classification via a Hierarchical Transformer

Model " [4]" addresses the challenge of long-term classification of food safety news events by proposing a new method that uses deep learning methods, specifically the BERT and Transformer models.. The paper opens with the disclosure importance of food safety and the growing media attention to this end Traditional models, such as BERT, are limited in the use of long documents, since they only process inputs up to 512 tokens, and lose important information in long texts In order to this will be overcome the authors devise a hierarchical model that divides long texts into blocks, processes it comes with a BERT sentence) and connects it to the Transformer model to attain efficient feature extraction by segmenting documents,. uses BERT for local content analysis, and the use of Transformers to capture global context, which is then used to classify food safety data into four categories: additives, content are heavy metals, dairy products, and counterfeits, 1999. It outperforms other commonly used models such as BERT-RNN, BERT-CNN, and FastText in terms of accuracy, recall, and F1 scores, especially excellence in classifying complex food safety classes Prescribed model has been developed through comparative tests on real-world data sets over time in the field of food safety -Positions itself as a state-of-the-art solution for data classification.

In "A Novel Foodborne Illness Detection Tool Based on Social Media" citetao2023novel, the author describes how social media, more precisely Twitter, is used to provide a foodborne illness outbreak detection. Observing the defects of standard surveillance systems, such as the National Outbreak Reporting System provided by CDC, which often suffer delays in timing, this present study introduces an alternative for improved real-time monitoring. In this paper, authors collect 430,000 geolocated tweets between 2017 and 2021 and apply deep machine learning models like BERTweet and RoBERTa to detect 110,000 tweets about foodborne illness. Data involving key entities such as foods, symptoms, and locations have been stored in a PostgreSQL database. Of course, resultant patterns from the Twitter data are very close to the ones at CDC, especially the implicated food categories. Descriptive and predictive analytics were conducted in the research to validate this approach, with its predictive models, including logistic regression at a level of 82% accuracy. The study demonstrated the potential that data from social media complement traditional surveillance systems. This new approach will move at a considerably faster pace and in real time regarding outbreaks of foodborne illness, and is likely to improve the timeliness and effectiveness of public health responses by reducing reliance on traditionally delayed data sources.

The paper ”Recurrent Neural Networks for Text Classification with Multitasking Learning”
citeliu2016recurrent discusses the limitations of traditional single-task learning models for the task of natural language processing (NLP) in that they lacked ample training data and suggested a multitasking learning algorithm using recurrent neural networks (RNNS). The three new models presented by the authors all share information between tasks through specific shared tasks, which enable using an integrated system training approach that utilizes shared knowledge role to enhance job-specific learning, raising at the same time the level of learning-related tasks. Four benchmark text classification data sets are tested on the model, and the results show a significant improvement in performance over single-task learning Shared-layer architecture, especially when fine-tuning, outperformed models the others, by an accuracy increase of 2.8%. Shared systems can better represent long-term references, whereas gating techniques enable the sharing of selective information, and various gating methods improve classification accuracy at large. In a word, the paper suggests multidisciplinary learning and improves the quality of knowledge sharing between related disciplines.

3 Problem Statement(s)

The aim of the **SemEval 2025 Task 9** is to move forward in explainable classification systems for detecting food hazards based on web-sourced food-incident report titles. Such systems could dramatically affect food safety by automatically identifying food-related issues from online sources, such as social media. This challenge's emphasis is on transparency because of the potential economic and health implications.

This task encompasses two main sub-tasks:

1. **Text Classification for Food Hazard Prediction (ST1):** Predict the type of hazards and the associated products from given food-incident report titles.

2. **Food Hazard and Product "Vector" Detection (ST2):** Predict the specific or the exact hazard and the product details, providing a detailed classification beyond a simple category prediction.

Participants will be evaluated on the basis of their system's ability to detect food hazards with a focus on the exact specification also. The evaluation metric is a two-step macro F1 score that prioritizes accuracy in hazard labeling for each sub-task.

4 New Suggestions / Novelty

- **Hierarchical Architectures**: Implement local classifiers for each node, organized by parent node and hierarchical level.

- **Hierarchical Attention Transformers**: Implement a Recursive hierarchy decoding approach within an encoder-decoder architeture to maintain hierarchial dependencies and each level awarness during text classification at each level.

- **Node-Level Models**: Utilize individual machine learning models at each node, such as:

  - Decision Trees
  - K-Nearest Neighbors (KNN)
  - Linear Regression
  - Logistic Regression

5 Plan for Implementation

5.1 Dataset Description

The SemEval 2025 Task 9 dataset consists of a total of 6,644 texts that are included in the SemEval 2025 Task 9 dataset, out of which 5,082 samples will be have to be assigned for training, 565 samples for the validation, and the remaining 997 as the test. The texts are between 5 and 277 characters in length, on average 88. Each entry of this dataset will then be described by features: year, month, day of recall issuance, language, country, title, and full text about recall description.

5.1.1 Class Imbalance

The dataset exhibits significant class imbalance in both the **Product-Category**, **Product**,**Hazard** and **Hazard-Category** features.

Product-Category In the case of the Product-Category, the most frequent category is "meat, egg, and dairy products", having 28.6% of all the recorded instances. Next comes "cereals and bakery products" with a 13.4% occurrence, and finally, "fruits and vegetables" at 10.7%.

Hazard-Category In the Hazard-Category, "allergens" are the most frequent occurence, having 36.8% of the total occurrences. This is followed by "biological hazards" at 34.5%, and "foreign bodies" at 11.1%. This imbalance presents notable challenges for classification tasks.

5.1.2 Features

- **Year:** Recall issuance year.
- **Month:** Recall issuance month.
- **Day:** Recall issuance day.
- **Language:** Primarily English.
- **Country:** Country of the recall.
- **Title:** Title of the food recall incident.

- **Text:** Full text of the recall description.

5.1.3 Labels

- **Product-Category:** There are 22 broader product categories with significant class imbalance.
- **Hazard-Category:** There are 10 broader hazard categories with notable imbalance.
- **Product:** 1,256 unique products.
- **Hazard:** 261 unique hazards.

5.1.4 Data Annotations

Each text is annotated by two experts in food science or food technology to ensure accuracy and reliability. The dataset is available under the Creative Commons BY-NC-SA 4.0 license.

5.1.5 Challenges and Evaluation

The significant class imbalance in both the hazard and product categories are required to be a careful handling to achieve a higher performance. The evaluation metric for this dataset is the macro F1 score, which assesses the accuracy of the hazard predictions across both sub-tasks (ST1 and ST2). Participants can utilize either the *title* or the *text* feature for their analysis.

5.2 Dataset Preprocessing

1. **Data Extraction**

   - **Merge Title and Text**: Combine the title and text fields into a single unified field for further processing.

2. **Text Preprocessing**

   - **Text Cleaning**:

     – **Remove Irrelevant Information**: Details that are of a non-essential nature, such as the case number, date and recall class, details in press releases, estimates of domestic, city, state, country, pounds recalled and recovery information should and have been excluded.

     – **Preserve Relevant Details**: Store names, product names, issues and descriptions must be kept in their correct way. For inquiries and responses, extract the following details: product defects; product hazards; product descriptions; problem identification in products; suggested consumer actions.

     – **Normalize Case**: Convert all text to lowercase for uniformity for the model.

     – **Eliminate Punctuation and Special Characters**: Remove punctuation marks and special symbols to simplify the text and removing un-necessary noise.

     – **Tokenization**: Break the cleaned text into the individual words and tokens for easier analysis.

     – **Remove Stop Words**: Exclude common and non-informative words that do not contribute significantly to any semantic meaning.

     – **Stemming/Lemmatization**: Reduce the words to their base or root forms to standardize the text and enhance consistency.

     – **Message Construction** : Formatted with a structured message approach to guide the model in extracting critical details linked to the product categories and hazard types while filtering out the irrelevant content.

     – **Model Utilization** : Leveraged the pre-trained Meta-Llama 3.2-3B-Instruct model to accurately extract key information from food incident reports, ensuring alignment with classification tasks and improving efficiency in identifying product hazards. This reduces the text size, making it more streamlined for the tasks.

  - **Efficient Output Generation** : Truncates the input to fit within the model's 1024-token limit, producing clean, structured outputs focused on essential information (e.g., product categories and hazard types) for easier classification and analysis.

3. **Label Encoding**

- **Convert Categorical Labels**:

  - **Numerical Encoding**: Transform categorical labels, such as hazard category, product category, hazard, and product, into numerical values for analytical purposes.

5.3 Model Implementation

5.3.1 Overview

To optimize our text classification results, we will focus on extensive experimentation. We will start with data preprocessing and utilize embeddings such as Bag of Words (BoW) and TF-IDF, treating each label—Hazard, Hazard-Category, Product, and Product-Category—independently. Next, we will explore advanced embeddings like MiniLM and implement transformer architectures including BERT, RoBERTa, and XLM-Net for fine-tuning.

5.3.2 Implementation Strategy

Data Preprocessing

- Clean and preprocess text data like how I have explained above.
- Split the dataset into training, validation, and test sets.

Initial Embedding Techniques

- **Bag of Words (BoW)**: Convert text into numerical vectors based on word frequencies.

- **TF-IDF**: Highlight important words using their frequency and inverse document frequency.
- **MP-Net Base V2**: A transformer-based embedding model that excels at capturing long-range dependencies and contextual relationships through its advanced permutation-based training objective, providing 768-dimensional embeddings for improved text analysis.

Machine Learning Models : Independently apply models to each label:

- **Decision Trees:** We implement the Decision Tree using the DecisionTreeClassifier from scikit-learn to perform the classification. The model is initialized with a fixed **random_state=42** to ensure reproducibility as same results are required each time. We used the **criterion=gini** which balances training speed and accuracy, and using the **splitter=best** for optimal splits which had to be done at each node at each levels.
- **K-Nearest Neighbors (KNN) :** We implement the K-Nearest Neighbors (KNN), a classification model using the KNeighborsClassifier from scikit-learn. The classifier is initialized with **n_neighbors=5**, specifying that the predictions will be based on the five nearest neighbors in the embedding space, the weights parameter we used **weights=distance** as it gives more weight to closer distanced neighbors.In the metric parameter we used the **metric=minkowski**.
- **Logistic Regression:** We implement Logistic Regression (LR), a linear classification model using the LogisticRegression class from scikit-learn. The classifier is initialized with **penalty=l2**, applying Ridge regularization to reduce overfitting. We specify **max_iter=200** to increase the number of iterations, ensuring the model converges. To handle class imbalances, we use the **class_weight=balanced**, which always automatically adjusts weights inversely proportional to the class frequencies. Finally, we set **random_state=42** for reproducibility as same results are required always and the use **multi_class=auto**, allowing the model to automatically choose among the appropriate strategy for multi-class problems
- **Support Vector Machine**: We implement the Support Vector Machine (SVM) classifier using the SVC class from scikit-learn.The classifier is initialized with **kernel=rbf**,

which uses a Gaussian distribution similar kernel to find a hyperplane that best separates the classes in the feature space. Additionally, we set **class_weight=balanced** it is specified to address class imbalance by adjusting the weights inversely proportional to the class frequencies. Finally, we put **random_state=42** to ensure reproducibility of the results.

- **Random Forest:** We implement RandomForestClassifier from scikit-learn. It operates by constructing the ensemble of decision trees, one among all trained on random subsets of the data, and then combines the predictions to improve the accuracy and reduces the overfitting. Key hyperparameters include **n_estimators=100**, which specifies the number of trees in the forest, and **random_state=42** to ensure reproducibility.

- **Naive Bayes:** We implement a Naive Bayes (NB) classifier using the MultinomialNB class from scikit-learn. The classifier is designed for categorical data and is particularly effective for text classification tasks. To handle label encoding, we use the **LabelEncoder** to transform the target variable into a numeric format. Before fitting the model, the features are normalized using **MinMaxScaler** to scale them within a fixed range, which helps improve model performance.

Evaluate performance using accuracy, precision, recall, and F1 score.

Advanced Embeddings : Implement MiniLM for improved context-based representations.

Transformer Architectures : Fine-tune BERT, RoBERTa, and XLM-Net for our classification task by using different methods like different layers of freezing/Un-Freezing along with using weighted loss to target imabalance in our dataset.

Hierarchical Architectures :

- **Creating the Hierarchy for Classifying Food Text Data**
  - **Embedding Generation**: Used the MPNet V2, we generated embeddings for four key components that are the classes: Product Category, Product, Hazard

Category, and Hazards. These embeddings captured the semantic relationships between labels, which will be used for the forming the basis for classification.

  - **Cosine Similarity for Relationship Mapping**: We computed the cosine similarity between the embeddings to measure the textual similarity in the high-dimensional space. This helped to identify the relationships and then the closeness between different labels, a critical step for hierarchical organization.
  - **Parent-Child Label Classification**: Labels were grouped into parent-child relationships:
    - **Parent Labels**: Labels with the highest similarity scores were classified as primary categories (e.g., Product Category and Hazard Category).
    - **Child Labels**: Labels with lower similarity scores were categorized under their respective parent labels, such as specific products under a product category, or hazards under a hazard category.
  - **Manual Review and Refinement**: A manual review process was conducted to identify and correct any misclassifications, ensuring accurate relationships between labels.
  - **Using GPT for Hierarchy Refinement**: To further improve the hierarchy:
    - Misclassifications were then identified by prompting GPT models with context about the intended structure we tried to make.
    - Adjustments to the hierarchy were suggested by the GPT and it was again checked manually to cross verify, especially for wiered and ambiguous cases where cosine similarity alone didn't correctly suffice and give proper explanations.

- **Improving Hierarchical Structure**
  - **Optimal Cluster Calculation**: The optimal number of clusters for each category is determined using two key methods:
    - **Elbow Method**: Identifies the "elbow point" in the inertia plot, suggesting the ideal cluster count by balancing inertia reduction with model complexity.
    - **Silhouette Score**: Evaluates cluster quality by measuring how well-separated and cohesive the clusters are, with higher scores indicating and explaining

better clustering.

  - **Clustering with Optimal K**: After selecting the optimal number of clusters, KMeans clustering is applied to group items based on their embeddings similarity, ensuring semantically similar items are clustered together.
  - **Outcome**: This approach enhances the hierarchical structure, which further improving the accuracy and coherence of item categorization.

- **Hierarchical Structures where Per Parent, Per Node and Each level were Local Classifiers are implemented.**
  - **Local Classifiers are choosen as SVM Models with the right set of Hyperparameters.**
- **Hierarchical Attention Transformers :** A encoder-decoder architecture where the encoder which is a simple RNN takes input text to produce a context matrix and the decoder, provided with hierarchial embeddings and level-wise masked self-attention, decodes this matrix into a sub-hierarchy sequence where level information as well as hierarchical dependencies are maintained. Then, the sequence is expanded level by level which enables to a parse tree by maintaining not just the ancestor-descendant relationships, but also the contextual importance of tokens.

6 Results and Analysis

Table 1: Performance Metrics (Macro F1 Score) for Different Machine Learning Models

Model	Hazard Category	Product Category	Hazard	Product
Logistic Regression	89	72	69	30
K Nearest Neighbours	86	76	68	41
Support Vector Machine	91	75	74	37
Decision Trees	68	45	42	17
Random Forest	84	70	66	39
Naive Bayes	86	68	73	46

SVM is consistently outperforming other models in terms of highest scores on Hazard Category (91%) and Hazard (74%) and also has a good score on Product Category (75%). KNN is at its best in the category of Product Category (76%) and given competitive results in Hazard Category (86%) and Hazard (68%).

Logistic Regression works well in Hazard Category (89%) and Hazard (69%), but is not impressive in the Product task (30%), with potential failure in coping with categorical data of different complexity. Naive Bayes is strongest in the Product task, scoring the highest (46%) with a really good score also in Hazard Category (86%) and Hazard (73%), thus pointing out its great fitness to categorical features tasks.

Random Forest performs almost equally good in all tasks, with decent scores on both Hazard Category (84%) and Product Category (70%). Decision Tree is the worst performing compared to the other techniques, having rather low scores across all tasks, which indicates a poor generalization ability on this data set.

SVM and KNN are the good choices for all tasks except for the Product classification task, which would be Naive Bayes. Decision Trees are a bad idea, since the Random Forest is performing decently.

Table 2: Performance Metrics (Macro F1 Score) for Different Transformer Models

Model	Hazard Category	Product Category	Hazard	Product
BERT	84	70	67	13
RoBerTa	81	72	61	11

While BERT outperforms RoBERTa on the Hazard Category and Hazard, respectively, at 84% and 67%, the scores in comparison to RoBERTa's 81% and 61%, a case can be made for BERT being better for hazard-related data due perhaps to the pretraining that captures better context-specific linguistic patterns. On the other hand, RoBERTa performs with a slight advantage in the Product Category, scoring 72% rather than BERT's 70%, probably because of the larger pretraining corpus and its better handling of nuances related to products.

As both struggle with Product classification, the scores were low (13% for BERT and 11% for RoBERTa), suggesting that the task might be particularly difficult, or the data is ambiguous. Although this also did not result in a strong improvement of RoBERTa over BERT, in the case of product-related classifications, the choice will likely be between RoBERTa and BERT as the latter is likely to be superior on hazard-related tasks.

Table 3: Performance Metrics (Macro F1 Score) for Different Hierarchical Models

Model	Hazard Category	Product Category	Hazard	Product
Classifier Per Parent Node (SVM)	91	74	75	37
Classifier Per Node (SVM)	89	73	74	39
Classifier Per Level (SVM)	88	68	73	20
Hierarchical Attention Transformer	28	22	19	9
OvR Classifier (SVM)	91	78	77	44

The table presents comparisons of Macro F1 scores across various hierarchical models belonging to different categories. Classifier Per Parent Node achieves the best score in Hazard Category at 91% and performed the poorest in Product at 37%, which emphasizes more favorable performance at higher levels of hierarchy. OvR Classifier posts a balanced performance with the highest at 91% in Hazard Category and the lowest at 44% in Product, thus generally well generalizing.

The Classifier Per Node (SVM) and Classifier Per Level (SVM) show a drop in performance as the hierarchy deepens, particularly in the Product category, where the Classifier

Per Level (SVM) scores just 20%. Overall, the OvR Classifier (SVM) appears the most versatile with consistent results across categories.

The Hierarchical Attention Transformer model gives extremely less results because there is an implementation bug in training, which was not able to be addressed yet, and also the parent paper which used Hierarchical Attention Transformers had the same attention mask and and context matrix was the same for all the parent and children relationship. This has to be addressed and changed. Coming to adding the Existing Ontology for Parent and children relationship, there is not much we could find which matched anything close to how the hierarchy of our dataset is defined.

References

[1] Zhe, Dong, Yujia Kang, Ruoqi Shao, and Zhenyi Liu. ‘A Food Safety Text Filtering Method Based on Text Classification Techniques.’ In *Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, 881–885. IEEE, 2020.

[2] Wang, Zhu, Booma Sowkarthiga Balasubramani, and Isabel F. Cruz. ‘Predictive Analytics Using Text Classification for Restaurant Inspections.’ In *Proceedings of the 3rd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics (UrbanGIS’17)*, 14. Association for Computing Machinery, 2017. doi:10.1145/3152178.3152192.

[3] Randl, Korbinian, John Pavlopoulos, Aron Henriksson, and Tony Lindgren. ‘CICLe: Conformal In-Context Learning for Large-Scale Multi-Class Food Risk Classification.’ 2024. arXiv:2403.11904. `https://arxiv.org/abs/2403.11904`.

[4] Xiong, Shufeng, Wenjie Tian, Vishwash Batra, Xiaobo Fan, Lei Xi, Hebing Liu, and Liangliang Liu. ‘Food Safety News Events Classification via a Hierarchical Transformer Model.’ *Heliyon* 9, no. 7 (2023). Elsevier.

[5] Tao, Dandan, Ruofan Hu, Dongyu Zhang, Jasmine Laber, Anne Lapsley, Timothy Kwan, Liam Rathke, Elke Rundensteiner, and Hao Feng. ‘A Novel Foodborne Illness Detection and Web Application Tool Based on Social Media.’ *Foods* 12, no. 14 (2023): 2769. doi:10.3390/foods12142769.

[6] Liu, Pengfei, Xipeng Qiu, and Xuanjing Huang. ‘Recurrent Neural Network for Text Classification with Multi-Task Learning.’ *arXiv preprint arXiv:1605.05101* (2016).