Anaselka at SemEval-2025 Task 9: Leveraging SVM and MNB for Detecting Food Hazard

Anwar Annas
National Research and Innovation Agency
Jakarta, Indonesia
anwa016@brin.go.id

Al Hafiz Akbar Maulana Siagian
National Research and Innovation Agency
Jakarta, Indonesia
alha001@brin.go.id

Abstract

This paper represents our participation in SemEval-2025 Task 9 focusing on food hazard detection challenge. In particular, we participate in Sub-task 1 of Task 9, that is, predicting "hazard-category" and "product-category" labels. To address this challenge, we leverages Support Vector Machine (SVM) and Multinomial Naive Bayes (MNB) in our submissions. We also utilizes Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and GridSearchCV in this work. Our obtained macro F1-Score results in the evalution phase are 0.6472 and 0.107 for SVM and MNB, respectively.

1 Introduction

Food safety is a critical public health concern that requires efficient monitoring and early detection of potential hazards (Fung et al., 2018). Food hazard detection challenge conducted in SemEval-2025 Task 9 is an important initiative aiming to develop explainable classification systems for detecting food safety issues based on textual data (Randl et al., 2025). This task is crucial because it can help automated crawlers identify and extract food-related incidents from sources like social media, which is crucial given the potential for significant economic impact. The challenge covers English-language food recall titles and is described in detail in the task overview paper of food hazard detection of SemEval-2025 (Randl et al., 2025).

Our system utilizes a multi-pronged approach to address the text classification task in Sub-task 1. First, we focus on robust text preprocessing of the dataset, such as removing special characters and numbers, converting text to lowercase, stemming, and excluding stop words. This preprocessing helps to standardize the input data and focus on the most relevant features (Kunilovskaya and Plum, 2021; Strasser and Klettke, 2024). For extracting features, we employ Bag of Words (BoW) (Salton and McGill, 1986) and Term Frequency-Inverse Document Frequency (TF-IDF) (Ramos et al., 2003) representations. These techniques capture the frequency and importance of key terms within the text providing informative input to our classification models.

To tackle the classification task, we leverage Support Vector Machines (SVM) (Cortes and Vapnik, 1995) as our classifier, while we use GridSearchCV for hyperparameter tuning (Bergstra and Bengio, 2012). The SVM classifier has the ability to handle high-dimensional feature spaces and identify optimal decision boundaries, which has proven to be a robust choice for this type of text classification problem. Furthermore, the use of GridSearchCV allows us to systematically explore a range of hyperparameter configurations, ultimately selecting the optimal settings for our specific task and dataset.

Participating in this challenge has provided valuable insights into the strengths and limitations of our system. Through an evaluation on the evaluation phase using test data, we are able to obtain an macro F1-score of 0.6858 in Sub-task 1. Although this obtained result might be categorized acceptable, we have identified areas where our system struggles, such as correctly classifying certain hazard and product categories that are less represented in the dataset.

2 Background

SemEval-2025 Task 9 is the food hazard detection challenge focused on developing explainable classification systems to identify food-related safety issues from textual data. The task involves predicting the type of hazard and product category given a textual input, such as the title of a food incident report. The input for this task consists of short English-language texts describing food recalls, with an average length of 88 characters. The dataset provided by the organizers includes 6,644 such texts, which were manually labeled by food

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 807–811
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

science and technology experts (Randl et al., 2025).

Task 9 consists of two sub-tasks. Sub-task 1's goal is to predict the "hazard-category" and "product-category" labels, while Sub-task 2's purpose is to predict the exact "hazard" and "product" values. Participants in the Task 9 can take part in both subtasks or they can choose to participate in one sub-task only. The Task 9's organizer (Randl et al., 2025) considered several phases, namely, a trial phase for model development, a conception phase for validation of unlabeled data, and an evaluation phase for final testing on labeled data. In this Task 9, we participate in Sub-task 1 only.

Related work in the area of explainable text classification for food safety risk detection is limited but growing. Recent studies have explored both model-specific (Assael et al., 2016) and model-agnostic (Ribeiro et al., 2016b,a) approaches to provide explanations for predictions. However, the unique challenges of this domain, such as the imbalanced class distribution and the need for precise hazard and product labels, present opportunities for novel contributions.

Our work aims to build upon these existing methods and address the specific requirements of the Sub-task 1 of SemEval-2025 Task 9. In particular, we strive to develop a robust and explainable system for detecting food safety risks from textual data by leveraging advanced text preprocessing techniques and powerful machine learning algorithms.

3 System Overview

Our system takes a multi-pronged approach to address the text classification task os Sub-task 1 in the SemEval 2025 Task 9. We focus on robust text preprocessing, effective feature extraction, and the use of powerful machine learning algorithms with hyperparameter optimization.

3.1 Data Preprocessing

The first step in our system's workflow is data preprocessing to standardize the input and focus on the most relevant features. We begin by removing special characters and numbers from the text, as these elements are often not directly relevant to the classification task. Next, we convert all text to lowercase to ensure consistency. To further enhance the quality of our features, we apply stemming using the Porter Stemmer, which reduces words to their base forms.

3.2 Feature Extraction

After the text preprocessing phase, we extract features from the cleaned text using two established techniques: Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF). The BoW approach captures the frequency of key terms within the text, providing a basic representation of the textual content. The BoW feature vector can be represented as:

$$XBoW = [f_1, f_2, ..., f_n]$$

The TF-IDF method, on the other hand, assigns higher weights to terms that are more important and distinctive within the corpus, further enhancing the informative nature of the feature representation. The TF-IDF value for a term t in a document d is calculated as:

$$TF - IDF(t, d) = TF(t, d) * IDF(t)$$

where TF(t, d) is the term frequency of t in d, and IDF(t) is the inverse document frequency of t in the entire corpus.

3.3 Classification

For the text classification task, we leverage Support Vector Machine (SVM) and Multinomial Naive Bayes (MNB) classifiers. Naive Bayes is a popular choice for text classification due to its simplicity, efficiency, and robustness. The Multinomial variant is particularly well-suited for discrete features, such as word counts, which is the case for our TF-IDF transformed text data.The Multinomial Naive Bayes classifier models the probability of a class c given the input features x as:

$$P(c|x) = (P(x|c) * P(c))/P(x) \quad (1)$$

where P(x|c) is the likelihood of the features given the class, P(c) is the prior probability of the class, and P(x) is the marginal probability of the features. Assuming the features are independent, the likelihood P(x|c) can be calculated as:

$$P(x|c) = P(x_i|c) \quad (2)$$

In the other hand, SVM is powerful machine learning algorithms that can effectively handle high-dimensional, sparse feature spaces, which is the case for our TF-IDF transformed text data (Cortes and Vapnik, 1995). The SVM method determines the best hyperplane to divide the classes by the

greatest amount. This is done by solving the following optimization problem:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \quad (3)$$

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \quad (4)$$

$$\xi_i \geq 0 \quad (5)$$

where w is the normal vector to the hyperplane, b is the bias term, ξ_i are the slack variables that allow for misclassifications, and C is the regularization parameter that controls the trade-off between the margin size and the number of misclassifications. We use the Linear SVC implementation from the scikit-learn library, which is suitable for large-scale linear classification tasks (Muppidi et al., 2021).

To optimize the performance of our classification models, we employ GridSearchCV, a technique that systematically explores a range of hyperparameter configurations and selects the optimal settings for our specific task and dataset. This approach allows us to fine-tune the SVM's hyperparameters, such as the regularization parameter (C) and the kernel function, to achieve the best possible classification results.

Our system aims to develop explainable and high-performing classification models for the Subtask 1 of SemEval-2025 Task 9, which is the food hazard detection challenge, by combining robust text preprocessing, effective feature extraction, and powerful machine learning algorithms with hyperparameter optimization.

4 Experimental Setup

We have set up our experimental workflow to run on Google Colab, a cloud-based Jupyter Notebook environment. Google Colab provides a free and accessible platform for running machine learning experiments, with access to GPU and TPU resources.

We utilized the provided train dataset containing a total of 5,082 samples. We did have not split the provided dataset into training, development, and test sets because the organizer also provided data for evaluation. To evaluate our trained models, we used a provided development dataset. The goal was to develop explainable and high-performing classification models for the Sub-task 1 of SemEval-2025 Task 9.

To prepare the input data for the classification models, we apply a series of text preprocessing steps as follows:

- Removal of special characters and numbers: We utilize regular expressions to remove all non-alphabetic characters from the text, leaving only the necessary words.
- Conversion to lowercase: All text is converted to lowercase to ensure consistency in the textual representation.
- Stemming using the Porter Stemmer: We employ the Porter Stemmer, a widely-used algorithm for reducing words to their base forms, to capture the semantic similarities between related terms.
- Stop word removal: We eliminate common words that do not carry significant meaning for the classification task, such as "the," "a," and "is," using the pre-defined list of English stop words from the NLTK library.

After the preprocessing stage, we extract features using two techniques as follows:

- Bag of Words (BoW): The BoW representation captures the frequency of key terms within the text, providing a basic representation of the textual content
- Term Frequency-Inverse Document Frequency (TF-IDF): This approach assigns higher weights to terms that are more important and distinctive within the corpus, further enhancing the informative nature of the feature representation.

We use BoW and TF-IDF to capture different aspects of the textual data and investigate which feature representation performs better for the classification task.

We use the training dataset to train and optimize the models. This will allow us to perform hyperparameter tuning and select the best-performing models. We leverage GridSearchCV from Scikit-learn, a technique that systematically explores a range of hyperparameter configurations and selects the optimal settings.

In the evaluation stages, the SemEval-2025 Task 9 focuses on the macro average F1-Score as an evaluation metric focusing on the hazard class (Randl et al., 2025)

.

	MNB	SVM
Macro average	0.1076	0.6858

Table 1: The obtained F1-Score of our models in the evaluation phase.

	Hazard	Product	Average
MNB	0.7903	0.4974	0.6439
SVM	0.9047	0.7322	0.8184

Table 2: The obtained F1-Score of our models on the training dataset.

5 Results

We submitted two models in the evaluation phase to participate in this Sub-task 1 of SemEval-2025 Task 9. In particular, our submission consisted of Multinomial Naive Bayes (MNB) and SVM models. Table 1 shows our obtained macro average F1-Score in the evaluation phase.

Our obtained results in Table 1 indicate that our SVM model could perform better than the MNB one. This SVM better performance corresponds to the obtained F1-Score of our models on the training dataset, as shown in Table 2. In particular, results in Table 2 shows that SVM could predict the hazard class correctly around 90%. On the other hand, results in Table 2 also demonstrates that MNB could not work as good as SVM, where the MNB predicted the hazard class correctly about 79% only. This disparity performance between SVM and MNB on the training dataset (Table 2) might affect the performance of our SVM and MNB models in the evaluation phase (Table 1) that focused on predicting the hazard class.

6 Limitations

The stark performance discrepancy of the Multinomial Naive Bayes (MNB) classifier—training macro F1-score of 0.64 versus evaluation macro F1-score of 0.10—stems from multiple interrelated factors. First, while hyperparameter tuning was rigorously applied during training via GridSearchCV to address initial low performance, an oversight led to the inadvertent use of default hyperparameters during evaluation. This inconsistency disrupted the model's calibrated probability estimates, amplifying its inherent sensitivity to imbalanced class distributions and sparse feature representations. MNB's reliance on term independence assumptions further clashed with the evaluation set's domain-specific contextual dependencies (e.g., multi-word hazards like "heavy metal contamination"), which BoW/TF-IDF failed to disentangle. The absence of optimized regularization during evaluation underscores the necessity of end-to-end hyperparameter consistency, particularly for models like MNB that lack intrinsic mechanisms to mitigate distribution shifts or lexical ambiguities in short, specialized texts.

7 Conclusion

Our system for the Sub-task 1 of SemEval-2025 Task 9 has been designed to tackle the complexities of identifying and categorizing food safety incidents from textual data. Through a rigorous experimental setup, we have developed a text classification solution that leveraged state-of-the-art techniques in data preprocessing, feature engineering, and model optimization. Our obtained submission results in the evaluation phase indicated that SVM could perform better than MNB. In particular, our SVM and MNB models achieved 0.6858 and 0.1076 of macro average F1-Scores, respectively. We assume this mediocre performance due to our models had difficulties in predicting the hazard class in the evaluation phase. For this reason, focusing on predicting the hazard class should be paid attention seriously in the future to deal with this challenging task.

References

Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas. 2016. Lipnet: End-to-end sentence-level lipreading.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Fred Fung, Huei-Shyong Wang, and Suresh Menon. 2018. Food safety in the 21st century. *Biomedical Journal*, 41(2):88–95.

Maria Kunilovskaya and Alistair Plum. 2021. Text preprocessing and its implications in a digital humanities project. In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, pages 85–93, Online. INCOMA Ltd.

Satish Muppidi, Balan Santhosh Kumar, and Korada Pavan Kumar. 2021. Sentiment analysis of citation sentences using machine learning techniques. In

2021 Innovations in Power and Advanced Computing Technologies (i-PACT), pages 1–5.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

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.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016a. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, San Diego, California. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016b. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery.

Gerard Salton and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., USA.

Sebastian Strasser and Meike Klettke. 2024. Transparent data preprocessing for machine learning. In *Proceedings of the 2024 Workshop on Human-In-the-Loop Data Analytics*, HILDA 24, page 1–6, New York, NY, USA. Association for Computing Machinery.