SRCB at SemEval-2025 Task 9: LLM Finetuning Approach based on External Attention Mechanism in The Food Hazard Detection

Yuming Zhang[1], **Hongyu Li**[1], **Yongwei Zhang**[1], **Shanshan Jiang**[1], **and Bin Dong**[1]

[1]Ricoh Software Research Center (Beijing) Co., Ltd
{Yuming.Zhang1, Hongyu.Li, Yongwei.Zhang,
Shanshan.Jiang, Bin.Dong}@cn.ricoh.com

Abstract

This paper reports on the performance of SRCB's system in SemEval-2025 Task 9: The Food Hazard Detection Challenge. We develop a system in the form of a pipeline consisting of two parts: 1. Candidate Recall Module, which selects the most probable correct labels from a large number of labels based on the BERT model; 2. LLM Prediction Module, which is used to generate the final prediction based on Large Language Models(LLM). Additionally, to address the issue of long prompts caused by an excessive number of labels, we propose a model architecture using the external attention mechanism to reduce resource consumption and improve performance. Our submission achieves the first place with the macro-F1 score of 54.73 on Sub-Task 2 and the third place with the macro-F1 score of 80.39 on Sub-Task 1. Our system is released at https://github.com/Doraxgui/Document_Attention.

1 Introduction

SemEval is a series of international research workshops aimed at advancing the field of natural language processing(NLP), with a particular focus on semantic analysis techniques and the creation of high-quality annotated datasets to address various complex challenges in natural language semantics. Each year, the workshop organizes a series of shared tasks, offering a platform for the presentation and comparative evaluation of computational semantic analysis systems developed by different teams. The Food Hazard Detection Task (Randl et al., 2025) comprises two sub-tasks: Sub-Task 1 focuses on classifying the type of hazard and product, while Sub-Task 2 aims to classify the exact hazard and product.

The challenges of this task include: 1) a heavily imbalanced class distribution, and 2) a large number of labels: 10 types of hazards and 22 types of products in Sub-Task 1, and 128 exact hazards and 1,142 exact products in Sub-Task 2. Constructing prompts with such a large number of labels for LLM-based response generation leads to high resource consumption and degraded performance.

For challenge 1, our proposed solution employs Large Language Models(LLMs). As LLMs demonstrate strong performance on data augmentation (Cai et al., 2023), particularly for imbalanced data. Therefore, we use LLM, specifically Qwen2.5-72B-Instruct (Yang et al., 2024), to perform data augmentation and deal with the imbalanced class distribution. For labels with limited training samples, we prompt the LLM to generate new samples based on the content of these samples, the semantic meaning of the label, and the text format of a randomly selected training sample. For challenge 2, we propose a novel model architecture named the **External Attention Mechanism**, which is used to combine input embeddings with label embeddings to inject label information into the input, as opposed to the conventional approach of concatenating label information in text form and relying on the self-attention mechanism. This innovative structure treats input and labels as two separate parts, reducing the length of prompts and leading to significant resource savings and performance improvements.

Our system is a pipeline consisting of two parts: **Candidate Recall Module** and **LLM Prediction Module**. The Candidate Recall Module is mainly composed of a BERT model with a classifier added at the top, which is used to filter all labels and keep those candidate labels which have high probability of correctness. The purpose of this module is to reduce the number of all candidate labels and help subsequent modules reduce error options and improve performance. The LLM Prediction Module first obtains the hidden states of the input and the hidden states of all labels, and then aligns them through the **External Attention**

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 996–1003

Mechanism (alternatively called Label Attention Module) to generate a new representation. Finally, the system uses this new representation to predict hazards and products in the form of Next Token Prediction task. The purpose of this module is to allow the hidden states of the input to calculate the attention mechanism with labels **externally**, rather than splicing labels directly into the input in the form of text and using Self-Attention Mechanism of LLM. In this way, the length of the LLM's input prompts can be significantly reduced. Moreover, in the External Attention Mechanism, all labels are treated equally, and the potential impact of the labels' order will not be introduced, as the hidden states of labels are calculated in parallel.

2 Background

The Food Hazard Detection Corpus, introduced in (Randl et al., 2025), includes the professional manually labeled titles and full texts on food recall collected from official food agency websites, and the language is English. Figure 1 shows an example of training data, our analysis reveals that relying solely on the title or text results in incomplete information extraction. Therefore, we combine both features to enhance the performance of the models.

(Edwards and Camacho-Collados, 2024) states that optimizing the top-layer classifier is ineffective for imbalanced class distribution. (Radford et al., 2019) introduces that the text generation mode based on Autoregressive LLM can be used as a better alternative method. (Plaza-del Arco et al., 2023) claims that LLM can adapt to different tasks without a large number of training samples due to their ability of understanding natural language instructions. Based on these studies, we focus mainly on LLM instruction tuning in our system, finetuning LLM to predict hazards and products.

3 Data

3.1 Data Processing

Our system concatenates the title and text into a single text format, followed by data cleaning. The cleaning process includes handling spaces and line breaks, removing duplicate natural segments, and converting characters into lowercase. Additionally, the system truncates the cleaned data to a specified maximum token length. For the Candidate Recall Module, the maximum token length is set to 512, while for the LLM Prediction Module, it is set to 1,024 (more than 90% of the training data tokens fall within this limit).

3.2 Data Augmentation

To address the class imbalance problem, we refer to the oversampling technique (Gosain and Sardana, 2017). For labels with fewer than 50 training samples (a threshold defined by ourselves), we use LLM, specifically the Qwen2.5-72B-Instruct, for data augmentation. Given a sample, we instruct the LLM to generate a new sample by combining the meaning of the given sample, the label of the given sample, and the writing style of a randomly selected sample from the training data. An example of our data augmentation is provided in A. Compared to repeated oversampling, this method is better aligned with the overall data distribution and enhances diversity.

4 System Description

For each classification task (including the classification of 1.type of hazard, 2.type of product, 3.exact hazard, 4.exact product), we use a unified pipeline consisting of two modules: **Candidate Recall Module** and **LLM Prediction Module**.

4.1 Candidate Recall Module

Our preliminary classification experiments with RoBERTa-base (Liu et al., 2019) indicate that while the macro-F1 score of the model is suboptimal, it achieves an impressive Recall score of over 95 for each label. This suggests that, given a sample, the BERT-like model will predict some candidate labels. These candidate labels: 1.always contain the gold label; 2.are much less than all labels, especially for Sub-Task 2, thereby mitigating the challenges posed by the large number of labels. In order to make full use of this feature, we add the Candidate Recall Module to the front of the LLM Prediction Module.

In the Candidate Recall Module, we adopt the one-vs-all approach (Galar et al., 2011), constructing a binary classifier on the top of BERT(Kenton and Toutanova, 2019) architecture for each label to predict the probability of a sample belonging to that label. For instance, we construct 128 classifiers for Sub-Task 2's exact hazards and 1,142 classifiers for Sub-Task 2's exact products. In our system, labels predicted by the classifier with a probability greater than 50% will be considered as

Input	title	Recall Notification: FSIS-024-94
	text	··· Product: SMOKED CHICKEN SAUSAGE ··· Problem: BACTERIA...
Sub-Task 1 output	hazard-category	biological
	product-category	meat, egg and dairy products
Sub-Task 2 output	hazard	listeria monocytogenes
	product	smoked sausage

Figure 1: Data sample

candidate labels. During training, negative samples are randomly selected from other labels in the training data. The Candidate Recall Module outputs a set of candidate labels. Compared to considering all labels, this approach significantly reduces the label space while maintaining a high probability of including gold labels.

4.2 LLM Prediction Module

In the LLM Prediction Module, candidate labels are incorporated into the processed data in the form of text. And our system needs to select a LLM as **Reference LLM** and another LLM as **Target LLM**. Reference LLM is used to process all labels, and Target LLM is used to be finetuned to process the input and generate the final prediction.

As depicted in Figure 2, the LLM Prediction Module is comprised of two modules: **Label Embedding Generation Module** and **Label Attention Module**. The Label Embedding Generation Module is used to generate the embeddings for all labels using Reference LLM, which can be considered as using Reference LLM to understand and generate the meaning of all labels. The Label Attention Module is designed to integrate the embeddings of processed data generated by Target LLM and the embeddings of all labels generated by Reference LLM. This module aligns the embeddings from Target LLM with those from Reference LLM, injecting all label information into the embeddings derived from Target LLM.

Due to the large number of labels, constructing prompts as a multiple choice task, where the LLM selects an option from a list candidate labels, is infeasible. Instead, we use External Attention Mechanism (alternatively called Label Attention Model) to combine the information of all labels while keeping the prompts concise, and we formulate the task as a Next Token Prediction task, where the LLM directly generates the label content. After instruction tuning, LLM can generate responses that can be parsed simply and mapped to those labels. Moreover, our designed structure is more suitable for traditional Next Token Prediction task, rather than the classification task.

4.2.1 Label Embedding Generation Module

Reference LLM generates embeddings (these embeddings are the hidden states which are used to map the entire LLM token vocabulary) for each token in all labels, which will be precomputed and stored to avoid real-time computation. The dimension of the hidden states (embeddings) would be (Length, Size), where **Length** is the number of tokens in the label, and **Size** is the size of hidden states. The hidden states of all labels are concatenated to form the ***Reference***, with dimensions (Number, MaxLength, Size), where **Number** is the total number of labels, **MaxLength** is the maximum token count across all labels, and **Size** is the size of hidden states. Zero-padding is applied for labels with fewer than MaxLength tokens.

For instance, for the exact hazards classification of Sub-Task 2, there are 128 labels, assuming that there are 5 tokens in each label. Input all the content of labels into Reference LLM and it will generate the ***Reference*** with the dimension of (128, 5, Size), where **Size** is the size of the hidden states. ***Reference*** will be stored locally for further calling, it contains hidden states (embeddings) of all tokens for all labels in one classification task.

The parameters of Reference LLM are frozen and the Reference LLM does not participate in training or inference, only the ***Reference*** will. And we choose Qwen2.5-14B-Base (Yang et al., 2024) as the Reference LLM.

This module aims to generate the understanding for all labels, which is stored as ***Reference***.

4.2.2 Label Attention Module

Given the processed data added with candidate labels, Target LLM generates its embeddings (the hidden states which are used to map the entire LLM token vocabulary), named as ***Query***. The dimension of ***Query*** is (QueryLength, Size), where

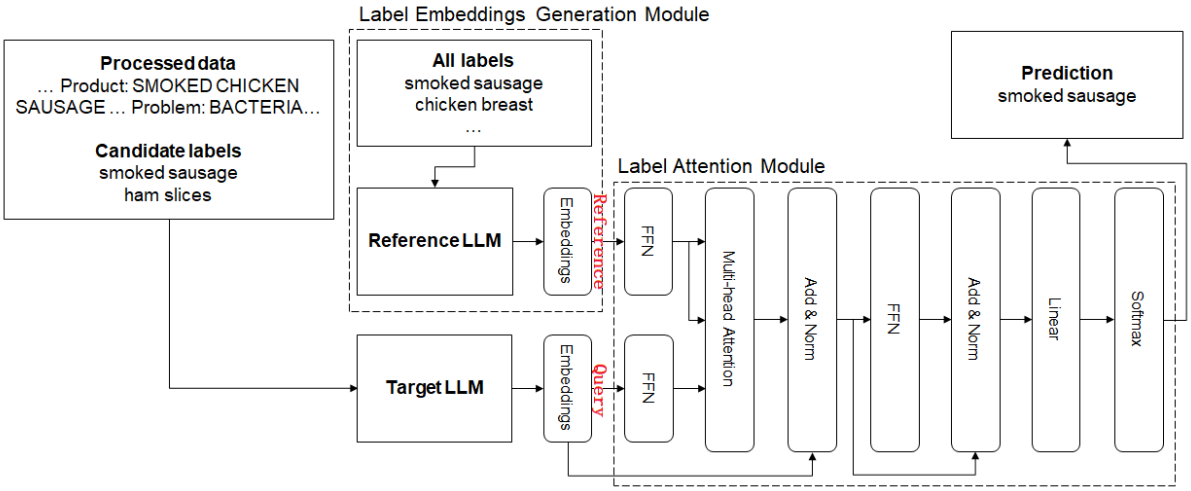

Figure 2: LLM Prediction Module

the **QueryLength** is the number of tokens in the processed data and **Size** is the size of hidden states.

For instance, if one processed data has 100 tokens, the dimension of ***Query*** would be (100, Size), where **Size** is the size of hidden states.

The parameters of Target LLM and the parameters in Label Attention Module need to be trained in training process. We choose Qwen2.5-14B-Base as the Target LLM (the same as the Reference LLM).

For alignment between ***Query*** and ***Reference***, we refer to the classic attention mechanism (Vaswani et al., 2017), treating ***Query*** as queries (Q) and ***Reference*** as keys (K) and values (V). We add a feedforward neural network (FFN) to each of them. In multi-head attention part, we set the number of attention heads to 12 and use the classic algorithm of attention mechanism as follow:

$$attention = Softmax(\frac{QK^T}{\sqrt{d_k}})V \qquad (1)$$

where **Q** is the result of FFN on ***Query*** and **K**, **V** are the results of FFN on ***Reference***. d_k is the dimension of **K**.

The output of multi-head attention exhibits significant deviation from the original Target LLM embeddings, which hinders model convergence if used directly in subsequent computations. Consequently, to reduce the burden of model training, we incorporate the embedding generated by Target LLM (***Query***) into the output of multi-head attention via a residual connection (He et al., 2016), followed by layer normalization (Ba et al., 2016) (the add&norm block). This approach consistently demonstrates significant performance improvements. Next, we simulate the structure of the Attention Mechanism from Transformers by incorporating an FFN, a residual connection followed by layer normalization. The Label Attention Module will output a new hidden state that has the same dimension with ***Query***.

During finetuning, we define the task objective as Next Token Prediction using the new hidden state of the output, rather than directly linear classification because text generation, through its inherent semantic understanding, can more effectively mitigate the impact of imbalanced class distribution. Specifically, when constructing the processed data, we add its gold label to the end of the data in the form of text. Because of the unidirectional attention mechanism of the decoder structure of LLM, it will not influence the calculation of External Attention Mechanism mentioned before. We add a linear layer to map the result embeddings combined by ***Query*** and ***Reference*** into the vocabulary space of LLM to predict the tokens of the gold label. When finetuning with the task objective of Next Token Prediction, we mask whole processed data except for the tokens of gold label we added at the end of the data, aiming to finetune the model only on the label prediction.

Furthermore, the Label Attention Module can resolve with the influence caused by the sequence of options. For instance, *prompt: Which one is better, A or B?* and *prompt: Which one is better, B or A?* generally have different answers, because of the inherent mechanism of LLM. However, the Label Attention Module addresses this is-

sue through matrix calculation. Unlike traditional approaches that rely on position embeddings, the matrix-based mechanism in the Label Attention Module does not involve explicit positional information, which treats each option fairly.

5 Experimental Setup

5.1 Data Splitting

For the Candidate Recall Module, we employ a 5-fold cross-validation, using 4 folds for training and the remaining fold for validation. For the LLM Prediction Module, due to the computing resource requirements (e.g. GPU memory and training time), we directly use the entire training dataset for training and the actual validation set for evaluation. Unlike smaller models such as BERT, LLM is less prone to overfitting during text generation tasks, which justifies this approach.

5.2 Hyperparameters

For the Candidate Recall Module, we select RoBERTa-base and DeBERTaV3-base (He et al., 2021) as base models. The learning rate is set to 1e-5 and the batch size is set to 4 with 4 negative samples. We use the AdamW optimizer and employ early stopping to prevent overfitting. For the LLM Prediction Module, we select Qwen2.5-14B-Base as the base model. The learning rate is set to 7e-6 and the effective batch size is 64 (achieved through gradient accumulation). We use the AdamW optimizer and train for 3 epochs.

5.3 Evaluation Measures

For Sub-Task 1, the measure is the average macro-F1 score of the type of hazard and product, which focuses more on the part of hazard. For Sub-Task 2, the measure calculates the average macro-F1 score of the exact of hazard and product, which also focuses on the part of hazard. In our system, the Candidate Recall Module is evaluated using the Recall score, which measures the ability to recall gold candidate label, counting whether the gold label is among the predicted candidate labels. The LLM Prediction Module is evaluated using the macro-F1 score for both hazard and product.

6 Results

6.1 Experiment Results

Candidate Recall Module The validation results on the cross-validation set are illustrated in Table 1, it indicates the average score of 5-fold cross-validation.

Both RoBERTa-base and DeBERTaV3-base achieve high Recall scores, exceeding 95, with DeBERTaV3-base slightly outperforming RoBERTa-base.

LLM Prediction Module The validation results on the actual validation set are illustrated in Table 2. All methods involve LLM predicting labels through text generation, differing in prompt construction and model structure. Method LLM inputs processed data without any labels into LLM. Method LLM+AL constructs prompts with processed data and all candidate labels. Method LLM+FL uses processed data and filtered candidate labels from Candidate Recall Module. Method LLM+FL+LPM incorporates LLM Prediction Module with processed data and filtered candidate labels.

Each value represents the macro-F1 score for each classification task, and the time column indicates the time consumption in hours. The reason why different methods show huge different time consumption is that some methods do not need a lot of memory, thus we increase their batch size and decrease their gradient accumulation during training. Memory consumption **mainly** depends on the length of prompt, therefore only the time consumption of Sub-Task 2 is evaluated (total amount of time consumption on exact hazard and exact product), as the time consumption of Sub-Task 1 is approximately the same.

Compared to LLM and LLM+AL, LLM+FL shows improvements in most tasks. With LPM, LLM+FL+LPM further enhances performance. Both using all labels, the time consumption of LLM+FL+LPM reduces compared to LLM+AL.

6.2 Test Results

We employ the majority voting approach for model ensemble, the differences between candidate models include adjustments to the LLM Prediction Module's structure, variations in the types of LLM and changes in Candidate Recall Module's models, and so on.

Our system achieves the highest macro-F1 score for Sub-Task 2 and a high score for Sub-Task 1. Table 2 details our result on the test set, showing performance consistent with the validation set except for the type of hazard, indicating potential overfitting.

	Sub-Task 1		Sub-Task 2	
PLM	type of hazard	type of product	exact hazard	exact product
RoBERTa-base	97.89	95.71	98.23	96.40
DeBERTaV3-base	**98.70**	**96.67**	**98.80**	**97.46**

Table 1: The recall scores on average validation (5-folds) set. PLM indicates Pretrained Language Model

Validation	Sub-Task 1		Sub-Task 2		
Method	type of hazard	type of product	exact hazard	exact product	time
LLM	80.47	75.19	68.52	38.14	**2**
LLM + AL	83.82	74.50	64.97	38.11	9
LLM + FL	82.44	81.78	67.64	39.64	4
LLM + FL + LPM	**89.81**	**82.59**	**69.62**	**40.39**	5.5
Test	Sub-Task 1		Sub-Task 2		
Method	type of hazard	type of product	exact hazard	exact product	time
LLM + FL + LPM	78.51	82.27	67.98	41.41	-

Table 2: The macro-F1 scores on validation and test set. LLM presents SFTed Qwen2.5-14B-Base. AL presents inputting all labels into prompt. FL presents inputting filtered labels from Candidate Recall Module into prompt. LPM presents LLM Prediction Module

6.3 Further Work

Several areas warrant further exploration and improvement: a) Optimizing the embedding generation method in the Label Embedding Generation Module. b) Using a different or larger Reference LLM in the LLM Prediction Module, with varying FFN sizes for mapping. c) Exploring the full use of the Label Attention Module by increasing the number of attention layers or adjusting the structure, and so on. In the future, we plan to explore and improve the LLM Prediction Module to achieve a more robust and efficient structure.

7 Conclusion

In this work, we propose a system consisting of the Candidate Recall Module and the Label Prediction Module. We identify that the primary cause of high training time consumption is the long prompt. To address this, we design the Candidate Recall Module to reduce the length of prompt in terms of system structure. And we design the Label Prediction Module to further minimize the impact of the prompt length using the External Attention Mechanism in terms of algorithm. Additionally, the Label Prediction Module also addresses the problem of the option sequence. With the help of other techniques like data augmentation and pipeline optimization, our system achieves the highest score for Sub-Task 2, and a competitive score for Sub-Task 1. For future work, we plan to explore advanced architectures to optimize the LLM Prediction Module.

8 Limitation

The LLM Prediction Module enhances performance and speed, but increases storage consumption. The results of the Label Embedding Generation Module are either generated in real time using the Reference LLM or pregenerated offline and stored locally, both imposing memory or storage burdens. Additionally, the LLM Prediction Module takes more time to train.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Xunxin Cai, Meng Xiao, Zhiyuan Ning, and Yuanchun Zhou. 2023. Resolving the imbalance issue in hierarchical disciplinary topic inference via llm-based data augmentation. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1424–1429. IEEE.

Aleksandra Edwards and Jose Camacho-Collados. 2024. Language models for text classification: Is in-context learning enough? *arXiv preprint arXiv:2403.17661*.

Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. 2011.

An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, 44(8):1761–1776.

Anjana Gosain and Saanchi Sardana. 2017. Handling class imbalance problem using oversampling techniques: A review. In *2017 international conference on advances in computing, communications and informatics (ICACCI)*, pages 79–85. IEEE.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1. Minneapolis, Minnesota.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Flor Miriam Plaza-del Arco, Debora Nozza, and Dirk Hovy. 2023. Wisdom of instruction-tuned language model crowds. exploring model label variation. *arXiv preprint arXiv:2307.12973*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

A Data Augmentation

Figure 3 presents the prompt we used to instruct LLM to generate new samples, and Figure 4 is an example of training data that needs to be augmented, while Figure 5 is a random sample selected from the training data. Figure 6 presents the data augmentation result, which has the similar format as ***Random sample*** and the same meaning as the ***Target sample***, we regard the ***Result*** as augmented data samples.

Prompt

<Reference Content>
<Title>
Target_Title
</Title>
<Text>
Target_Text
</Text>

hazard: ***Target_Label***
</Reference Content>

<Reference Style>
<Title>
Random_Title
</Title>
<Text>
Random_Text
</Text>

hazard: ***Random_Label***
</Reference Style>

Please refer to the meaning of hazard in <Reference Content> and the writing format in <Reference Style> to create a new document for antibiotics, vet drugs, which should be different from <Reference Content>.

Figure 3: Prompt for data augmentation

Target sample

Target_Title:
Recall Notification: FSIS-026-94

Target_Text:
Case Number: 026-94
Date Opened: 07/12/1994
Date Closed: 09/19/1994

Recall Class: N
Press Release (Y/N): N
Domestic Est. Number: N/A
Name: MILI QUALITY FOODS
Imported Product (Y/N): Y
Foreign Estab. Number: 47
City: ASHRAT
State: F
Country: ISRAEL
Product: CHICKEN BREAST, COOKED
Problem: DRUG
Description: SULFADIMETHOXINE
Total Pounds Recalled: 8,260
Pounds Recovered: 0

Target_Label:
antibiotics, vet drugs

Figure 4: Target sample

Random sample

Random_Title
Jindi Cheese – potential microbial contamination (Listeria spp) – 18 January 2013

Random_Text:
Page Content Lactalis Jindi Pty Ltd has recalled the following product nationally due to potential microbial contamination (Listeria spp). Date notified to FSANZ: 18/01/2013 Food type: Dairy products Product brand names: Aida Valley Blue Cow Coles Finest Dynasty Emporium Enterprize G&K Harris Farm International Jindi Jindi Reserve Kenilworth Kingaroy Old Telegraph Road Raw Materials Siena Tomewin Farm Tomme Farm Top Paddock Wattle Valley Willow Grove Package description and size: Various sizes Various packages foil wrap - gold and silver, shrink wrapped, cardboard box, white paper wrapped Date marking: Various Best Before dates Country of origin: Australia Reason for recall: potential microbial contamination (Listeria spp) Consumer advice: Consumers should not eat these products. Any consumers concerned about their health should seek medical advice Contact: Lactalis Jindi Pty Ltd 1800 680 175 Further information Jindi product images (zip file 1105kb) Jindi full product list (updated 22 January 2013) Food recall on soft cheese - media release Related links:

Random_Label:
microbiological contamination

Figure 5: Random sample

Result

<Title>
Mili Quality Foods – Potential Antibiotic Contamination (Sulfadimethoxine) – 12 July 1994
</Title>
<Text>
Page Content Mili Quality Foods has recalled the following product due to potential antibiotic contamination (Sulfadimethoxine). Date notified to FSIS: 12/07/1994 Food type: Poultry Product brand name: Mili Quality Foods Chicken Breast Package description and size: Various sizes, typically 5-10 lb (2.27-4.54 kg) packages Date marking: Various Best Before dates Country of origin: Israel Reason for recall: potential antibiotic contamination (Sulfadimethoxine) Consumer advice: Consumers should not consume these products. Any consumers who have experienced adverse reactions after consuming the product should seek medical advice. Contact: Mili Quality Foods Customer Service 1-800-123-4567 Further information Recall details and product images (zip file 500kb) Related links:
</Text>

hazard: antibiotics, vet drugs

Figure 6: Result