Conformal Prompting for Largescale Multiclass Food Risk Classification

Anonymous ACL submission

Abstract

Contaminated or adulterated food poses a substantial risk to human health. Given sets of labeled web texts for training, Machine Learning and Natural Language Processing can be applied to automatically extract pointers towards such risks in order to generate early warnings. We publish a dataset of 7,619 short texts describing food recalls. Each text is manually labeled, on two granularity levels (coarse and fine), for food products and hazards that the recall corresponds to. We describe the dataset, also presenting baseline scores of naive, traditional, and transformer models. We show that Support Vector Machines based on a Bag of Words representation outperform RoBERTa and XLM-R on classes with low support. We also apply in-context learning with PaLM, leveraging Conformal Prediction to improve it by reducing the number of classes used to select the few-shots. We call this method Conformal Prompting.

1 Introduction

Food-bourne illnesses and contaminated food pose a serious threat to human health and lead to thousands of deaths (Majowicz et al., 2014; de Noordhout et al., 2014). Natural Language Processing (NLP) solutions based on Machine and Deep Learning (ML, DL) or Large Language Models (LLMs) may enable fast responses to new threats by generating warnings from publicly available texts on the internet. These texts, however, as we show, can be noisy and are characterized by thousands of classes. While we know about the existence of singular text-based datasets on the topic of food-bourne illnesses (Hu et al., 2022), they focus on the detection of such illnesses rather than their classification. In this work, we address this topic and provide the following contributions:

1. We present the first (to our knowledge) text-based dataset for food hazard classification comprising labels for food product and food hazard classification, at two granularities.[1]

2. We present a benchmark on the introduced dataset, using naive, traditional ML, and DL-based classifiers, showing that a Support Vector Machine outperforms the rest due to its better performance for low-support classes.

3. We propose a method based on Conformal Prediction (Vovk et al., 2005, CP) to improve the shot selection for in-context learning (ICL) of LLMs, enabling not only feasible but also more accurate prompting.

In the following sections, we first present some related work in Section 2 and then describe our methods, data, and results in Sections 3 and 4. Lastly, we discuss the results and conclude this paper in Section 5 followed by a short discussion of limitations to our approach.

2 Background and Related Work

The formulation of the attention mechanism by Vaswani et al. (2017) has drastically changed the field of NLP. ML models based on this architecture and the self-supervised masked language modeling pre-training paradigm easily outperform traditional methods both on monolingual (Devlin et al., 2019; Liu et al., 2019) and multilingual (Conneau et al., 2020) data. As training transformers usually takes enormous amounts of data, already pre-trained models are typically further trained (*fine-tuned*) on smaller amounts of task-specific data. Although transformers were originally described as an encoder-decoder architecture (Vaswani et al., 2017), mapping an input-text to an output-text, models intended for classification tasks usually only employ an encoder (Devlin et al., 2019; Liu

[1] Publicly available, hidden for anonymity.

et al., 2019; Conneau et al., 2020), that is followed by a number of task-specific layers (called the *head*). These heads are usually added to the pre-trained base model directly before *fine-tuning*.

Recently, LLMs such as PaLM (Chowdhery et al., 2022) have been shown to exceed the capabilities of smaller transformers even without further fine-tuning. That is, by simply providing a context of a few labeled samples per class, LLMs are able to predict the classes of unseen samples within this context. As LLMs are usually text-to-text transformers, the context is provided directly in each prompt. This paradigm is commonly referred to as *few-shot-prompting* or *in-context learning* (Brown et al., 2020). Since the detection of LLMs' few-shot capabilities, many newer LLMs have been designed for high few-shot performance (Gao et al., 2021; Chowdhery et al., 2022).

Current work on few-shot prompt engineering focuses mostly on creating/finding the optimal few-shot samples from the training data rather than limiting the number of classes these samples are taken from. Ahmed et al. (2023) proposed a workflow for automatically finding similar samples from the pool of labeled "training" samples for code summarization. Shi et al. (2023) focused on automatic generation of Chain of Thought (CoT) labels for samples in reasoning tasks. CoT is another prompting paradigm that asks the LLM to provide a chain of reasoning before delivering the prediction and has been shown to drastically improve reasoning performance (Wei et al., 2022). Nevertheless, to the best of our knowledge, there is no previous work on how to leverage few-shot-prompting in LLMs for multiclass prediction problems with 100+ classes.

The CP framework (Vovk et al., 2005) is a methodology for associating predictions of a classification algorithm with confidence guarantees, e.g., if a user in a multi-class setting wants to be sure that 95% of the predictions are correct, CP will do this by outputting a set of labels, i.e., multi-label adhering to the set guarantees. It can be applied to any classification (or regression) algorithm, as long as a calibration set is set aside together with a nonconformity function. In our context, the stronger the classification algorithm, and the better the nonconformity function, the fewer labels (hence, shots) will be produced with the same guarantee. We point to the work of Vovk et al. (2005); Johansson et al. (2014); Bostrom et al. (2017) for more on CP.

3 Method

3.1 Problem Description

The basic problem we address in this paper is extreme multi-label classification on heavily imbalanced data. More formally, given a number of training-texts $T_i,\ i \in \{1, 2, \ldots, N\}$ and their vector of corresponding classes Y_i we aim to train a classifier $\mathrm{f}(T_i, \cdot) = \hat{Y}_i$ that minimizes the error $|Y_i - \hat{Y}_i|$. For standard ML classifiers, the function $\mathrm{f}(T_i, \cdot)$ is usually a two step process with the first step mapping the text to a machine readable embedding vector $X_i = \mathrm{e}(T_i)$, and the second one involving the learning process on X_i: $\mathrm{f}(T_i, \cdot) = \mathrm{f}'(X_i, \cdot)$. For the class-labels in $Y_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,M}]$ we have $y_{i,j} \in \{0, 1\} \forall i, j$, while $\hat{Y}_i = [\hat{y}_{i,1}, \hat{y}_{i,2}, \ldots, \hat{y}_{i,M}]$ is the vector of probabilities of T_i belonging to class j and therefore we have $\hat{y}_{i,j} \in [0, 1] \forall i, j$. For all our classes, we have at least $M > 10$. While only in rare cases, our data contains samples with $|Y_i| > 1$, i.e. multiple true class-labels (multilabel).

3.2 ML Classifiers

As a naive baseline, we report the performance of two classifiers. The random classifier (RANDOM) yields each $\hat{Y}_i$ by generating a random number in $[0, 1]$ for each $\hat{y}_{i,j}$. The support-based baseline (SUPPORT), chooses each $\hat{y}_{i,j}$ to be the normalized support of the class:

$$\hat{y}_{i,j} = \frac{\sum_{k=1}^{N} y_{k,j}}{N}.$$

In order to select a set of predicted classes from the probabilities in $\hat{Y}_i$, we first amplify the differences in our models' predictions per sample, by normalizing them according to

$$\frac{\hat{Y}_i - \max_{1 \le j \le M}(\hat{y}_{i,j})}{\max_{1 \le j \le M}(\hat{y}_{i,j}) - \min_{1 \le j \le M}(\hat{y}_{i,j})}.$$

Afterwards, we assume all classes j with $\hat{y}_{i,j} > 0.5$ to be predicted by the model.

3.2.1 Traditional ML Classifiers

We use Bag-of-Words (**BOW**) and Term Frequency - Inverse Document Frequency (Spärck Jones, 1972, **TF-IDF**) encodings, combined with LR or SVM classifiers.[2] During the training of the SVMs we use a linear kernel and optimize the parameter

[2]For the classifiers, we use the implementation from the Python library `scikit-learn` (Pedregosa et al., 2011).

$C \in \{0.5, 1.0, 2.0\}$ for L2 regularization on the validation splits. For LR we use a 'liblinear' solver and optimize $C \in \{0.5, 1.0, 2.0\}$ for both L1 and L2 regularization. For both classifiers, we use a one-vs-all approach to multilabel classification, which means that we train one binary classifier for each class in the predicted label.

Text pre-processing comprises the application of a `TreebankWordTokenizer`, followed by `PorterStemmer` from the `nltk` (Bird et al., 2009) Python package. For the BOW representation,[3] we create a vector:

$$X_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,V}], \; \forall x_{i,v} \in [0, 1]$$

for each sample i, where V is the vocabulary size, and $x_{i,v}$ is the normalized count of occurrences of token v in sample i. For the TF-IDF embedding, for each sample i we calculate a vector:

$$X_i = \sum_{j=0}^{M_i} [\mathrm{tf}_{j,1} \cdot \mathrm{idf}_j, \mathrm{tf}_{j,2} \cdot \mathrm{idf}_j, \ldots, \mathrm{tf}_{j,N} \cdot \mathrm{idf}_j]$$

where N is the number of samples in the training data, $\mathrm{tf}_{j,k}$ is the count of the samples' j-th token in the k-th training sample, and $\mathrm{idf}_j = \ln \frac{N}{n_j}$ with n_j being the number of documents that contain token j. In order to make the above embeddings independent of the sample length we normalize X_i with the L2-norm in both cases.

3.2.2 Encoder-only Transformers

As a more recent counterpart to the previously described traditional ML classifiers, we fine-tune two models from huggingface's `transformers`[4] library: RoBERTa$_{base}$ (Liu et al., 2019) and XLM-RoBERTa$_{base}$ (Conneau et al., 2020, XLM-R) in their `base`-sizes (RoBERTa: *125M params*; XLM-R: *270M params*). Both models use the structure introduced by BERT$_{base}$ (Devlin et al., 2019, *L=12, H=768, A=12*), which improves comparability of their results. The different parameter counts result mainly from the different vocabulary sizes used in their Byte-Pair-based encoders (Sennrich et al., 2016): RoBERTa uses a vocabulary of size 50k, while XLM-R uses 250k tokens. For the purposes of this paper, the most important difference between the models is that while RoBERTa is only pre-trained on English texts, XLM-R is pre-trained on 100 different languages.

To fine-tune these two models, we use the standard sequence classification heads provided by the `transformers` library. We optimize training using AdamW (Loshchilov and Hutter, 2019) in combination with a learning rate that stays constant for the first two epochs and then declines linearly towards 1% of its starting value after 20 epochs. Furthermore, we employ early stopping with a patience of four epochs on the maximum macro F_1 score computed on the validation set. We also use the validation data to optimize the learning rate over the values $2 \cdot 10^{-5}$, $3 \cdot 10^{-5}$, and $5 \cdot 10^{-5}$. Due to hardware limitations, we only use a batch size of 16 samples for training.

3.2.3 Prompting

As a final classifier, we employ few-shot prompting with PaLM (Chowdhery et al., 2022), a text-to-text transformer with 540B parameters. We use three different kinds of prompts: (**PaLM-ALL**) A context describing the classification task followed by two randomly ordered text-label pairs taken from the training data per class; (**PaLM-CONF**) This is the basis of *Conformal Prompting*, where the context describes the classification task, followed by two labeled texts per class, ordered most probable first, in a reduced prediction set of length n_{conf} derived by CP (Vovk et al., 2005); (**PaLM-LIMIT**) A context describing the classification task followed by the first n_{conf} samples of the PaLM-ALL prompt. The purpose of the less intuitive PaLM-LIMIT prompt is to provide a baseline for PaLM-CONF. An example of each of these prompts is shown in Appendix A.

CP uses the concept of a non-conformity measure $\delta(Y_i, \hat{Y}_i)$ in order to create sets of predicted classes, which contain the true class with a probability of $p \geq 1 - \alpha$. In our case, this non-conformity measure is simply the error on the true class: $\delta(Y_i, \hat{Y}_i) = 1 - \hat{y}_{i,j} | y_{i,j} = 1$. Here, $\hat{Y}_i$ are the predictions of the best traditional classifier (BOW-SVM), scaled so that $\sum_{j=0}^{M} \hat{y}_{i,j} = 1$, and $\hat{Y}_i$ are the corresponding true labels. If one defines $q = \frac{\lceil (N+1)(1-\alpha) \rceil}{N}$, and $\hat{q}$ as the qth empirical quantile of $\{\delta(Y_i, \hat{Y}_i) | 1 \leq i \leq N\}$, it can be shown that for any prediction $\hat{Y}$ of the classifier on an unknown sample, a set of classes $\{j | \hat{y}_j \geq 1 - \hat{q}\}$ contains the true class with probability $p \geq 1 - \alpha$ (Vovk et al., 2005). In this paper, we focus on the pure few-shot performance of PaLM. Therefore, we decided not to employ additional prompt-engineering

[3] We use or own implementation for the representations.
[4] https://huggingface.co/docs/transformers/index

techniques, such as CoT or leveraging sample similarity. Our code is publicly available (under a CC BY-NC-SA 4.0 license) at (*hidden for anonymity*).

4 Empirical Analysis

In the following section, we first introduce our dataset and, following that, present the classification performance of the methods introduced above.

4.1 Data Description

The dataset consists of 7,619 short texts (length in characters: min=5, avg=84, max=360), which are the titles of food-recall announcements (therefore referred to as "`title`"), crawled from 24 food-recall domains (governmental & NGO, see Table 1) by Agroknow.[5] The texts are written in 6 languages, with English ($n = 6,713$) and German ($n = 892$) being the most common ones, followed by French ($n = 8$), Greek ($n = 4$), Italian ($n = 1$) and Danish ($n = 1$). As shown in Figure 1, most of the texts have been authored after 2010. The texts describe recalls of specific food products due to specific reasons. Each of the texts has been assigned six labels encoding these foodstuffs and hazards:

1. `hazard`: A fine-grained description of the hazards mentioned in the texts comprising 409 classes.

2. `hazard_category`: A categorized version of the `hazard` label comprising 11 classes.

3. `hazard_title`: A collection of character spans, generated from the LR feature importance. These are signifying parts of the `title` important for the hazard classification.

4. `product`: A fine-grained description of the products mentioned in the texts comprising 1,901 classes.

5. `product_category`: A categorized version of the `product` label comprising 29 classes.

6. `product_title`: A collection of character spans, generated from the LR feature importance. These are signifying parts of the `title` important for the product classification.

The dataset, publicly released under a Creative Commons BY-NC-SA 4.0 license, comprises also metadata, such as the release date of the text (columns `year`, `month`, and `day`), the language of the text (column `language`), and the country of issue (column `country`).

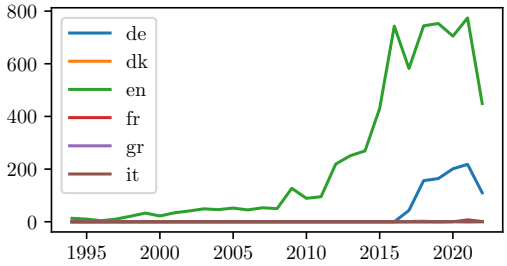


Figure 1: Languages in the dataset per year.

Domain	Samples
www.fda.gov	1760
www.fsis.usda.gov	1131
www.productsafety.gov.au	928
www.food.gov.uk	914
www.lebensmittelwarnung.de	890
www.inspection.gc.ca	864
www.fsai.ie	365
www.foodstandards.gov.au	282
inspection.canada.ca	126
www.CPs.gov.hk	123
recalls-rappels.canada.ca	101
tna.europarchive.org	52
wayback.archive-it.org	23
healthycanadians.gc.ca	18
www.sfa.gov.sg	11
www.collectionscanada.gc.ca	10
securite-alimentaire.public.lu	8
portal.efet.gr	4
www.foodstandards.gov.scot	3
www.ages.at	2
www.accessdata.fda.gov	1
webarchive.nationalarchives.gov.uk	1
www.salute.gov.it	1
www.foedevarestyrelsen.dk	1

Table 1: Data sources, ordered by support number

Quantifying the Noise in the Data

It is important to note that samples were labeled not only based on the `title`, but also the content of the food recall article. This means that some of the samples will not contain evidence for all the classes assigned to them. As irregular samples are common in real-world data, we decided to not filter the dataset for such samples, but rather provide a measure for the noise of the data in the `hazard_title` and `product_title` labels. In order to produce this measure, we

[5] https://agroknow.com/

use the coefficients of the BOW-LR classifiers for the hazard_category and product_category classes to extract important terms per class.

For each text-label pair T_i, Y_i, we split T_i in tokens $\{t_{i,1}, t_{i,2}, \ldots, t_{i,K}\}$, using the process described in Section 3.2.1. Afterward, we calculate a score by adding the positive model coefficients associated with $t_{i,k}$ if $y_{i,j} = 1$, and subtracting the positive model coefficients associated with $t_{i,k}$ if $y_{i,j} = 0$. Although the quality of these terms depends on class support, they can still help us frame the noise in the data by focusing on informative tokens, i.e., tokens with a positive coefficient for a specific class. We find that each such token corresponds to 1.28 classes on average for the hazard_category and 1.62 classes for the product_category.[6] Also, we see that 14.86% of the terms for hazard_title and 26.78% for product_title are empty, indicating that evidence for the class is missing in the titles. A few sample texts are shown in Appendix B.

Ground Truth

The labels were assigned by one human Agroknow curator per web-domain. Additionally, randomized checks of the labels were performed by more experienced curators. In the unlikely case of disagreement between the experts, the label assigned by the second more experienced curator is retained.

Class Imbalance

One of the most prominent features of the data is the heavy class imbalance. Figure 2 shows the sample counts per class and label. All the labels in the data show a long-tail distribution, with just a small number of classes having most of the samples. Therefore, we extract sets of high-support classes $\mathcal{C}_{high}$ and low-support classes $\mathcal{C}_{low}$ comprised of around one third of the total number of samples in the data for each label. The classes included in these sets are highlighted by a grey background in Figure 2. For the hazard_category label, the $\mathcal{C}_{high}$ is comprised of only one class with 2579 samples, and the $\mathcal{C}_{low}$ consists of nine classes with 2487 samples in total, and for hazard $|\mathcal{C}_{high}| = 2654$ samples in 3 classes and $|\mathcal{C}_{low}| = 2490$ samples in 392 classes. For product_category we have $|\mathcal{C}_{high}| = 2852$ samples in 3 classes and $|\mathcal{C}_{low}| = 2297$ samples in 21 classes, and for product $|\mathcal{C}_{high}| = 2538$ samples in 73 classes and $|\mathcal{C}_{low}| = 2528$ samples in 1522 classes.

[6] Estimation for the fine-grained labels is difficult because of low per-class support.

4.2 Training

In order to train and evaluate our ML-models, we apply 5-fold Cross-Validation (CV) to create 5 train-test splits. From each of these 5 training sets, we create a validation set using 10% holdout. For both of these splitting techniques, we use stratification on the hazard_category label as this is the label with the least number of classes and therefore provides a sufficient number of samples for splitting in each class. We keep the same splits for all labels, in order to keep the results comparable. This implies that, for labels other than hazard_category, the standard deviation over the splits may be higher. Our classifiers present baseline performance on the dataset. In order to demonstrate the effect of class imbalance on performance, we do not employ balancing methods like oversampling or class weights during training.

4.3 Experimental Results

In this section, we present the predictive performance of the classifiers described in Section 3.

The ML Baselines

The classification scores presented in Table 2 clearly show that all the classifiers outperform both naive baselines. The overall best-performing classifier for all four labels is a simple BOW-SVM model. Nevertheless, when only looking at the high-support classes, the best position is usually taken by one of the encoder-only transformers. The scores on the low-support classes show that BOW-SVM's strength is creating acceptable classification performance with very few training samples. In this segment, BOW-SVM massively outperforms the transformers, which need a relatively high number of samples to achieve good results even with transfer learning. This theory is supported by the much lower relative performance of RoBERTa and XLM-R on labels with more than 100 classes (i.e. hazard and product), as for these the number of classes with less than 100 samples is much higher.

Transformers

The good performance of RoBERTa and XLM-R on high-support classes in the hazard label can be explained by the relatively high number of samples per class in this segment compared to the product label (see Figure 2). Surprisingly, the multilingual XLM-R only outperforms RoBERTa in the high-

Model	Scores (*all classes*)		Scores ($\mathcal{C}_{high}$)		Scores ($\mathcal{C}_{low}$)	
	F_1 (*macro*)	Accuracy	F_1 (*macro*)	Accuracy	F_1 (*macro*)	Accuracy
`hazard_category`						11 *classes;* $\alpha = 0.05$
RANDOM	0.13 ± 0.00	0.00 ± 0.00	0.47 ± 0.01	0.48 ± 0.01	0.06 ± 0.00	0.00 ± 0.00
SUPPORT	0.09 ± 0.00	0.00 ± 0.00	0.25 ± 0.00	0.34 ± 0.00	0.00 ± 0.00	0.67 ± 0.00
BOW-LR	0.46 ± 0.02	0.68 ± 0.01	0.81 ± 0.01	0.82 ± 0.01	0.38 ± 0.03	0.76 ± 0.01
BOW-SVM	$\mathbf{0.52 \pm 0.03}$	$\mathbf{0.73 \pm 0.02}$	0.85 ± 0.01	0.86 ± 0.01	$\mathbf{0.46 \pm 0.04}$	$\mathbf{0.80 \pm 0.01}$
TF-IDF-LR	0.40 ± 0.02	0.65 ± 0.01	0.78 ± 0.01	0.79 ± 0.01	0.32 ± 0.02	0.75 ± 0.01
TF-IDF-SVM	0.47 ± 0.04	0.70 ± 0.01	0.83 ± 0.01	0.83 ± 0.01	0.39 ± 0.05	0.78 ± 0.01
RoBERTa	0.47 ± 0.02	$\mathbf{0.73 \pm 0.04}$	0.87 ± 0.03	0.88 ± 0.03	0.39 ± 0.03	0.78 ± 0.03
XLM-R	0.45 ± 0.03	0.72 ± 0.02	$\mathbf{0.89 \pm 0.02}$	$\mathbf{0.90 \pm 0.02}$	0.36 ± 0.04	0.78 ± 0.02
PaLM-ALL	0.44 ± 0.04	0.60 ± 0.01	0.84 ± 0.01	0.87 ± 0.01	0.36 ± 0.05	0.76 ± 0.01
PaLM-LIMIT	0.27 ± 0.04	0.21 ± 0.01	0.61 ± 0.01	0.73 ± 0.01	0.26 ± 0.05	0.70 ± 0.01
PaLM-CONF	0.45 ± 0.03	0.68 ± 0.01	0.86 ± 0.01	0.88 ± 0.01	0.37 ± 0.03	0.77 ± 0.01
`hazard`						409 *classes;* $\alpha = 0.20$
RANDOM	0.00 ± 0.00	0.00 ± 0.00	0.18 ± 0.00	0.12 ± 0.01	0.00 ± 0.00	0.00 ± 0.00
SUPPORT	0.00 ± 0.00	0.00 ± 0.00	0.21 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.67 ± 0.01
BOW-LR	0.09 ± 0.01	0.43 ± 0.01	0.52 ± 0.02	0.61 ± 0.01	0.08 ± 0.01	0.71 ± 0.01
BOW-SVM	$\mathbf{0.11 \pm 0.01}$	$\mathbf{0.46 \pm 0.01}$	0.52 ± 0.03	$\mathbf{0.62 \pm 0.02}$	$\mathbf{0.10 \pm 0.01}$	$\mathbf{0.72 \pm 0.01}$
TF-IDF-LR	0.05 ± 0.01	0.35 ± 0.02	0.47 ± 0.01	0.48 ± 0.02	0.04 ± 0.01	0.69 ± 0.02
TF-IDF-SVM	0.08 ± 0.01	0.42 ± 0.01	0.49 ± 0.01	0.56 ± 0.01	0.07 ± 0.01	0.70 ± 0.01
RoBERTa	0.03 ± 0.00	0.20 ± 0.04	$\mathbf{0.54 \pm 0.03}$	0.58 ± 0.03	0.02 ± 0.00	0.30 ± 0.04
XLM-R	0.02 ± 0.00	0.19 ± 0.02	0.53 ± 0.03	0.58 ± 0.04	0.01 ± 0.00	0.25 ± 0.05
PaLM-LIMIT	$\mathbf{0.14 \pm 0.01}$	0.39 ± 0.01	0.66 ± 0.02	0.81 ± 0.01	$\mathbf{0.12 \pm 0.01}$	$\mathbf{0.55 \pm 0.02}$
PaLM-CONF	$\mathbf{0.14 \pm 0.01}$	0.39 ± 0.01	$\mathbf{0.67 \pm 0.02}$	$\mathbf{0.82 \pm 0.01}$	$\mathbf{0.12 \pm 0.01}$	0.54 ± 0.02
`product_category`						29 *classes;* $\alpha = 0.05$
RANDOM	0.06 ± 0.00	0.00 ± 0.00	0.20 ± 0.01	0.13 ± 0.01	0.03 ± 0.00	0.00 ± 0.00
SUPPORT	0.03 ± 0.00	0.00 ± 0.00	0.22 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.70 ± 0.01
BOW-LR	0.49 ± 0.01	0.56 ± 0.01	0.61 ± 0.02	0.70 ± 0.02	0.44 ± 0.02	0.82 ± 0.01
BOW-SVM	$\mathbf{0.54 \pm 0.02}$	0.62 ± 0.01	0.66 ± 0.01	0.75 ± 0.01	$\mathbf{0.50 \pm 0.03}$	$\mathbf{0.83 \pm 0.01}$
TF-IDF-LR	0.38 ± 0.02	0.42 ± 0.01	0.50 ± 0.02	0.55 ± 0.02	0.32 ± 0.02	0.78 ± 0.00
TF-IDF-SVM	0.47 ± 0.02	0.54 ± 0.01	0.58 ± 0.01	0.66 ± 0.01	0.43 ± 0.03	0.81 ± 0.01
RoBERTa	0.50 ± 0.02	$\mathbf{0.63 \pm 0.01}$	0.72 ± 0.03	0.79 ± 0.02	0.43 ± 0.02	0.80 ± 0.01
XLM-R	0.47 ± 0.01	0.61 ± 0.03	$\mathbf{0.73 \pm 0.03}$	$\mathbf{0.80 \pm 0.02}$	0.39 ± 0.01	0.77 ± 0.02
PaLM-LIMIT	0.41 ± 0.05	0.34 ± 0.05	0.41 ± 0.05	0.69 ± 0.01	0.40 ± 0.05	0.74 ± 0.01
PaLM-CONF	$\mathbf{0.58 \pm 0.01}$	$\mathbf{0.66 \pm 0.01}$	$\mathbf{0.74 \pm 0.02}$	$\mathbf{0.84 \pm 0.01}$	$\mathbf{0.53 \pm 0.01}$	0.82 ± 0.01
`product`						1901 *classes;* $\alpha = .5$
RANDOM	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
SUPPORT	0.00 ± 0.00	0.03 ± 0.00	0.00 ± 0.00	0.03 ± 0.00	0.00 ± 0.00	0.67 ± 0.01
BOW-LR	0.06 ± 0.00	0.25 ± 0.01	$\mathbf{0.32 \pm 0.01}$	0.42 ± 0.01	0.01 ± 0.00	0.64 ± 0.01
BOW-SVM	$\mathbf{0.07 \pm 0.00}$	$\mathbf{0.27 \pm 0.01}$	0.30 ± 0.01	$\mathbf{0.46 \pm 0.01}$	$\mathbf{0.02 \pm 0.00}$	0.62 ± 0.01
TF-IDF-LR	0.02 ± 0.00	0.15 ± 0.01	0.16 ± 0.01	0.20 ± 0.01	0.00 ± 0.00	$\mathbf{0.66 \pm 0.01}$
TF-IDF-SVM	0.04 ± 0.00	0.20 ± 0.01	0.21 ± 0.01	0.33 ± 0.02	0.01 ± 0.00	0.58 ± 0.05
RoBERTa	0.00 ± 0.00	0.00 ± 0.00	0.05 ± 0.00	0.04 ± 0.01	0.00 ± 0.00	0.13 ± 0.05
XLM-R	0.00 ± 0.00	0.01 ± 0.01	0.02 ± 0.00	0.02 ± 0.01	0.00 ± 0.00	0.15 ± 0.02
PaLM-LIMIT	$\mathbf{0.12 \pm 0.00}$	0.20 ± 0.01	$\mathbf{0.48 \pm 0.03}$	$\mathbf{0.57 \pm 0.02}$	$\mathbf{0.05 \pm 0.01}$	0.59 ± 0.01
PaLM-CONF	$\mathbf{0.12 \pm 0.00}$	0.20 ± 0.01	$\mathbf{0.48 \pm 0.02}$	$\mathbf{0.57 \pm 0.01}$	$\mathbf{0.05 \pm 0.01}$	0.59 ± 0.01

Table 2: Average model performance and standard deviation over 5 CV-splits. Bold scores are the best score per column and label.

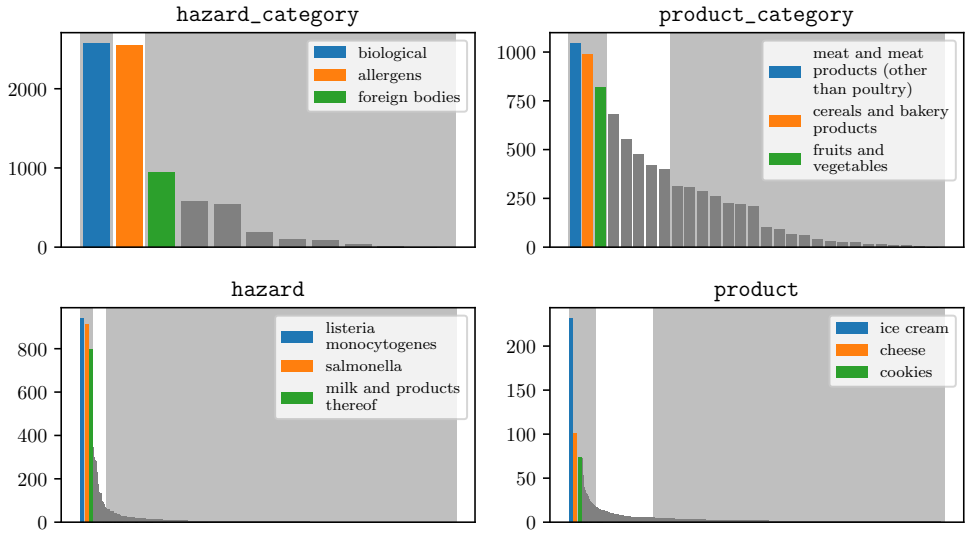


Figure 2: Class support with background grayed for high- (on the left) and low-support classes (right) for performance analysis. For reasons of better readability, we only name the three most supported classes per task.

support segments of the `hazard_category` and `product_category` labels even though the texts come in multiple languages. We assume the low number of per-class samples is not sufficient for the very large embedding layer of XLM-R.

Prompting

The naive approach to prompting (PaLM-ALL) performs below the average of the non-naive classifiers on the `hazard_category` label on all segments. As in prompting, the predicted class is delivered in free text, it is possible that the LLM produces output that is not within the set of class labels. While these outputs can sometimes be interpreted as belonging to one of the class labels, we only count exact matches. In case of the `hazard_category`, prompting failed to predict any class for 22% of the samples (average over the CV splits).

4.4 Conformal Prompting

Few-shot prompting (e.g., with two samples per class) is not feasible for the `hazard`, `product_category`, and `product` labels (even for `product_category`, prediction takes around 4 s per sample) due to the high number of classes. In order to reduce the number of classes for which we include shots in the prompt, we leverage CP to yield *Conformal Prompting* with the PaLM-CONF model. CP utilizes a classifier's certainty on each of the predictions in order to build sets of predicted classes that statistically contain the true class with a previously specified probability. As can be seen in Figure 3, CP leads to more concise prediction sets than just taking the k classes the classifier is most certain about (referred to as "max-k"). In contrast to max-k, CP produces shorter sets if the classifier has a high certainty on the true class. Nevertheless, classifiers with less security will improve on max-k's accuracy only at larger sets.

	PaLM-LIMIT		PaLM-CONF	
	size	*fails*	*size*	*fails*
`hazard_category`	776	67%	830	9%
`product_category`	2043	40%	2098	5%
`hazard`	2324	23%	2368	15%
`product`	2838	13%	2882	12%

Table 3: Prompt length (avg) in characters and cases (%) where prompting failed to deliver a valid class label.

As we are aiming for a set size below 40 (i.e., 20 classes, two shots per class), we select the α based on Figure 3. For `hazard_category` and `product_category`, we choose $\alpha = 0.05$ (meaning $p = 0.95$), as even for this high accuracy we get set sizes smaller than 10. For the remaining labels we are limited by set length and choose $\alpha = 0.2$ for `hazard` and $\alpha = .5$ for `product`. This means, that for `product` we have to accept a chance of at most 50% that the prediction set does not contain the ground truth in order to keep the set size low.

Table 3 presents the (sensitive to a) average prompt length and prompting failure rates. While the prompts are of comparable size (we attribute the slight increase in PaLM-CONF to the added statement on example order, see Appendix A), the LLM produces more valid class labels when using CP to create the samples. For `hazard_category` the prompt lengths are much lower than for PaLM-

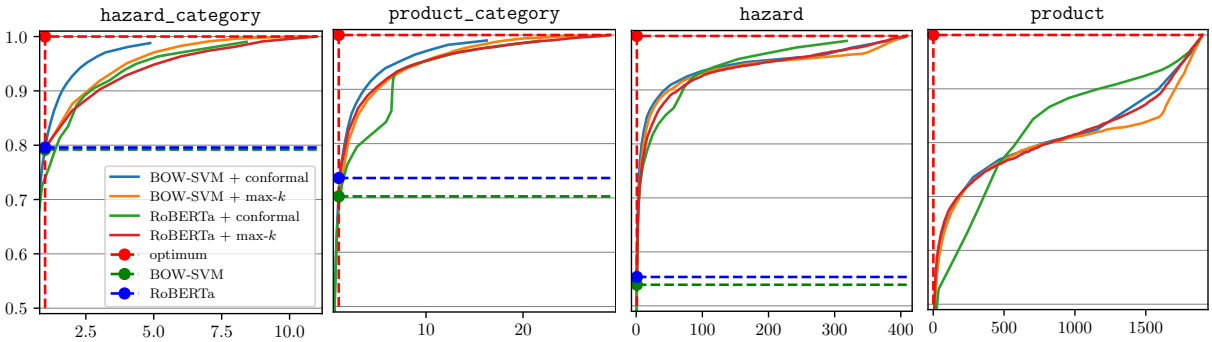


Figure 3: CP performance on the best traditional classifier and the best encoder-only transformer. The optimal prediction set (red dashed line) would have one element which is the true class with a probability of 1. y**-axis:** probability of the true class being in the prediction set; x**-axis:** length of the prediction sets

ALL (2,227 characters). The failure rate of PaLM-CONF is also reduced compared to the 22% fails in PaLM-ALL. Intuitively, the failure rate of PaLM-LIMIT is decreasing with increasing prompt length (more samples lead to fewer failures).

Taking a closer look at the performance of PaLM-LIMIT and PaLM-CONF (see Table 2), we see that although PaLM-CONF underperforms compared to BOW-SVM in `hazard_category`, it still outperforms both prompting baselines on all metrics. For a higher number of classes, the performance of PaLM-CONF drastically increases compared to the other models, making it the best tested classifier on the `hazard`, `product_category`, and `product` labels in terms of F_1. This increase can be seen in both high-support and low-support classes, although performance on $\mathcal{C}_{high}$ shows a bigger increase compared to BOW-SVM. We attribute the difference in performance between PaLM-LIMIT and PaLM-CONF decreases for higher classes to the higher α in these cases: sacrificing CP-guarantees for faster prediction.

5 Discussion & Conclusions

We present a novel dataset for multi-label classification of short texts describing food recalls. The dataset contains expert-annotated labels on food products and hazards on two levels of granularity, coarse (tens) and fine (hundreds). Additionally, the dataset includes computer-generated spans highlighting possible evidence for the class labels. While these spans are by no means perfect, they can give an estimate of the noise in the labels. We present baseline performance (F_1-score and accuracy) for naive, traditional, and deep classifiers for all four classification tasks in the proposed dataset. The dataset is publicly available under a CC BY-NC-SA 4.0 license at (*hidden for anonymity*).

Additionally, we show that reducing the number of few-shot examples for prompting with PaLM, by only taking into account classes from a conformal set, reduces prediction time while at the same time increasing performance in terms of F_1. Our results suggest that a simple random reduction of few-shot examples (reflected in the PaLM-LIMIT baseline) already makes prompting a strong approach compared to our other methods. In this setting, we leverage the well-documented strong in-context reasoning capabilities of LLMs (Chowdhery et al., 2022; Wei et al., 2022; Shi et al., 2023), which do not necessarily need a full view of all the possible classes. Nevertheless, we show that if the examples are taken from a reduced population that is very likely to contain the true class, we can further improve on this performance even without using other prompting techniques such as example matching (Ahmed et al., 2023) or CoT.

Our results suggest that Conformal Prompting (at $\alpha < 0.1$) outperforms normal prompting. Depending on the data and number of possible classes, this may or may not extend to other classifiers (Table 2; `hazard_category`, `product_category`). Although it allows PaLM to more accurately predict the class label (even from $\alpha \leq 0.5$), Conformal Prompting is also subject to a trade-off between predictive and temporal performance, i.e., sacrificing accuracy (i.e., due to hardware and time constraints) when more classes are involved, yet outperforming other traditional and deep classifiers, and leading to fewer prompting failures than random few-shot sampling. This shows that Conformal Prompting is a promising approach for large-scale multiclass classification with LLMs, whose applications to other domains and datasets should be further explored in future work.

6 Limitations

Nevertheless, the dataset and approach discussed in this paper are subject to limitations. Regarding the dataset, we identified the following:

- The labels in our dataset are subject to noise. Specifically, some samples are missing evidence for one or more of the assigned classes, while tokens important for classification may indicate more than one class. This may lead to classifiers trained on the data seeing contradicting examples and therefore limit their predictive performance.

- The spans in `hazard_title` and `product_title` are machine generated and not manually curated. This means that while they give an estimation of word importance, they are no gold standard for explainability tasks.

We aim to improve these limitations in future iterations of the dataset. For our approach leveraging CP for few-shot prompting, we found the following limitations:

- As visualized in Figure 3, CP represents a trade-off between high prediction set accuracy and low set length relative to the total number of classes. This means that with a rising total number of classes, we will have to sacrifice predictive performance in order to keep the prompt size feasible, which ultimativelly might render the approach useless.

- In this paper we used normal conformal prediction, which only guarantees a certain probability of a single true class being in the prediction set. In order to achieve true multilabel guarantees we would need to switch to mondrian CP.

- We only verify our approach on a single LLM. This means that our approach might not be generalizable to other LLMs and perform differently or not at all for few-shot prompting with such models.

While we have to accept the first of these points as inherent to the approach, we are planning to address the remaining points in future work. As all the data used in our dataset was already publicly available before the publication of this work, we do not violate anybody's privacy by republishing it.

Acknowledgements

This work was funded by the European Union.

References

Toufique Ahmed, Kunal Suresh Pai, Premkumar Devanbu, and Earl T. Barr. 2023. Improving few-shot prompts with relevant static analysis products.

S. Bird, E. Loper, and Klein E. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Henrik Bostrom, Lars Asker, Ram Gurung, Isak Karlsson, Tony Lindgren, and Panagiotis Papapetrou. 2017. Conformal prediction using random survival forests. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 812–817.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Charline Maertens de Noordhout, Brecht Devleesschauwer, Frederick J Angulo, Geert Verbeke, Juanita Haagsma, Martyn Kirk, Arie Havelaar, and Niko Speybroeck. 2014. The global burden of listeriosis: a systematic review and meta-analysis. *The Lancet Infectious Diseases*, 14(11):1073–1082.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners.

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

Ulf Johansson, Henrik Boström, Tuve Löfström, and Henrik Linusson. 2014. Regression conformal prediction with random forests. *Machine learning*, 97:155–176.

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.

I. Loshchilov and F. Hutter. 2019. In *7th International Conference on Learning Representations, ICLR 2019*, 7th International Conference on Learning Representations, ICLR 2019, University of Freiburg.

Shannon E. Majowicz, Elaine Scallan, Andria Jones-Bitton, Jan M. Sargeant, Jackie Stapleton, Frederick J. Angulo, Derrick H. Yeung, and Martyn D. Kirk. 2014. Global incidence of human shiga toxin–producing escherichia coli infections and deaths: A systematic review and knowledge synthesis. *Foodborne Pathogens and Disease*, 11(6):447–455. PMID: 24750096.

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.

R. Sennrich, B. Haddow, and A. Birch. 2016. Neural machine translation of rare words with subword units. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, volume 3, pages 1715–1725 – 1725, School of Informatics, University of Edinburgh.

Fobo Shi, Peijun Qing, Dong Yang, Nan Wang, Youbo Lei, Haonan Lu, and Xiaodong Lin. 2023. Prompt space optimizing few-shot reasoning success with large language models.

K. Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Appendix

A Sample Prompts:

Sample prompts taken from the `hazard_category`-label follow. Examples occurring in multiple prompts are colour-coded for better visibility. Observe the addition in PaLM-CONF, stating the ordering of the samples (underlined). The order of the examples in PaLM-ALL and PaLM-LIMIT is identical, while in PaLM-CONF they are sorted by model certainty (most probable one first). In PaLM-LIMIT, we avoid getting double examples per class unless the number of examples in the prompt is greater than the total number of unique classes.

PaLM-ALL:

Context start:

We are looking for food hazards in texts. Here are some labelled examples:

"Evermore Group Pty Ltd – New Choice Assorted Jelly Cups" -> food additives and flavourings

"Milbona Gouda jung gerieben, mindestens 7 Wochen gereift, 250 g" -> foreign bodies

"Yekta Foods recalls Achachi Jelly Cups because of a choking hazard" -> other hazard

"Schweppes Lemon Lime and Bitters" -> fraud

"Recall of Sriracha Hot Chili Sauce due to Risk of the Contents Exploding" -> packaging defect

"Recall of a Batch of Global Botanics CBD Paste Due to the Presence of Unsafe Levels of Delta-9-tetrahydrocannabinol (THC)" -> chemical

"Mundella Foods-Feta Supreme Mediterranean Style Feta Cheese" -> biological

"NulaCPoods Pty Ltd-No Udder Coconut Yoghurt, Alpine Coconut Yoghurt Natural, Alpine Coconut Yoghurt Passionfruit" -> allergens

"Kraft Heinz Foods Company Recalls Turkey Bacon Products Due To Possible Adulteration" -> organoleptic aspects

"Bamboo Aroma Sip Cup" -> migration

"Sunshine Sprouts – Alfalfa Sprouts" -> biological

"Thuan Phat Supermarket Croydon Park – New Choice Milky Pudding Jelly and taro jelly cups" -> food additives and flavourings

"coles mini classics ice creams" -> foreign bodies

"Wilderness Family Naturals brand Coconut Milk Powder and Coconut Chia Pudding Mix recalled due to undeclared milk" -> allergens

"IGA-Christmas Kisses (Cream filled sponge cakes)" -> fraud

"PepsiCo recalls Tropicana Trop 50 Multivitamins Juice" -> organoleptic aspects

"USA LESS Issues Voluntary Nationwide Recall of LEOPARD Miracle Honey Due to Presence of Undeclared Sildenafil" -> chemical

"Silikon-Muffinbackform" -> migration

"Deutscher Winzerglühwein, weiss, 0,75 L" -> packaging defect

"Two Brothers Pork Skins Recalls Pork Skin Products Due to Misbranding and Failure to Produce Under A HACCP Plan" -> other hazard

Context end:

Please predict the correct class for the following sample:

"Frickenschmidt Foods LLC Recalls Ready-to-Eat Beef Stick Products Due to Misbranding" ->

PaLM-LIMIT:

Context start:

We are looking for food hazards in texts. Here are some labelled examples:

"Evermore Group Pty Ltd – New Choice Assorted Jelly Cups" -> food additives and flavourings

"Milbona Gouda jung gerieben, mindestens 7 Wochen gereift, 250 g" -> foreign bodies

"Yekta Foods recalls Achachi Jelly Cups because of a choking hazard" -> other hazard

"Schweppes Lemon Lime and Bitters" -> fraud

"Recall of Sriracha Hot Chili Sauce due to Risk of the Contents Exploding" -> packaging defect

"Recall of a Batch of Global Botanics CBD Paste Due to the Presence of Unsafe Levels of Delta-9-tetrahydrocannabinol (THC)" -> chemical

"Mundella Foods-Feta Supreme Mediterranean Style Feta Cheese" -> biological

"NulaCPoods Pty Ltd–No Udder Coconut Yoghurt, Alpine Coconut Yoghurt Natural, Alpine Coconut Yoghurt Passionfruit" -> allergens

Context end:

Please predict the correct class for the following sample:

"Frickenschmidt Foods LLC Recalls Ready-to-Eat Beef Stick Products Due to Misbranding" ->

PaLM-CONF:

Context start:

We are looking for food hazards in texts. Here are some labelled examples sorted from most probable to least probable:

"Schweppes Lemon Lime and Bitters" -> fraud

"IGA-Christmas Kisses (Cream filled sponge cakes)" -> fraud

"Milbona Gouda jung gerieben, mindestens 7 Wochen gereift, 250 g" -> foreign bodies

"coles mini classics ice creams" -> foreign bodies

"NulaCPoods Pty Ltd–No Udder Coconut Yoghurt, Alpine Coconut Yoghurt Natural, Alpine Coconut Yoghurt Passionfruit" -> allergens

"Wilderness Family Naturals brand Coconut Milk Powder and Coconut Chia Pudding Mix recalled due to undeclared milk" -> allergens

"Recall of a Batch of Global Botanics CBD Paste Due to the Presence of Unsafe Levels of Delta-9-tetrahydrocannabinol (THC)" -> chemical

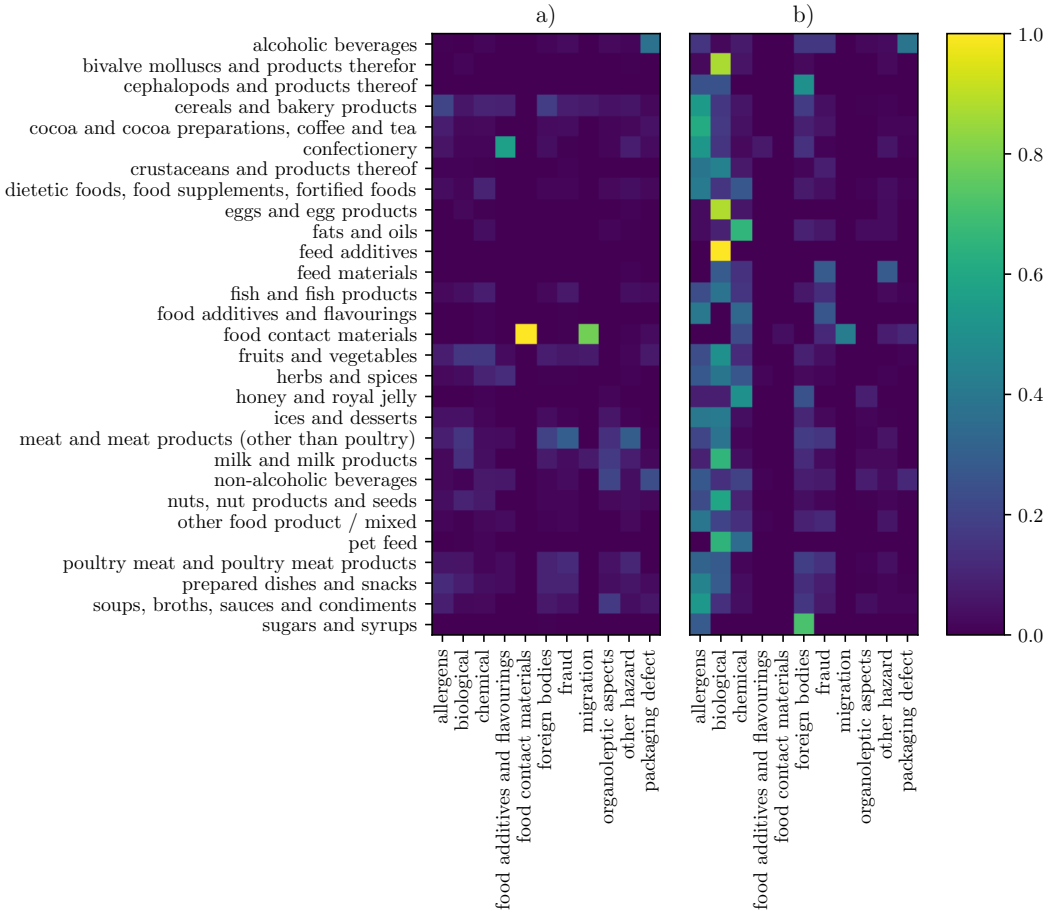
"USA LESS Issues Voluntary Nationwide Recall of LEOPARD Miracle Honey Due to Presence of Undeclared Sildenafil" -> chemical

Context end:

Please predict the correct class for the following sample:

"Frickenschmidt Foods LLC Recalls Ready-to-Eat Beef Stick Products Due to Misbranding" ->

B Data sample:



Label co-occurence for the `hazard_category` and `product_category` classes normalized by **a)** `hazard_category`, and **b)** `product_category`. While this is not true for all the `hazard_category-product_category` pairs, some show strong linkage.

Some labeled sample texts. Colored spans signify the spans in `hazard_title` and `product_title`:

"Butterball LLC Recalls Turkey Products Due to Possible Salmonella Schwarzengrund Contamination"				
Labels	`hazard`:	salmonella schwarzengrund	`hazard_category`:	biological
	`product`:	fresh minced turkey	`product_category`:	poultry meat and poultry meat products

"2009 - peanut corporation of america announces voluntary nationwide recall of peanut butter"				
Labels	`hazard`:	salmonella	`hazard_category`:	biological
	`product`:	peanut butter	`product_category`:	nuts, nut products and seeds

"V&S Imports and Exports Pty Ltd — Yayla Natural Yoghurt and Try Me Natural Yoghurt"				
Labels	`hazard`:	escherichia coli	`hazard_category`:	biological
	`product`:	yoghurt	`product_category`:	milk and milk products

"Undeclared Wheat , Egg , Milk and Soya in O'Dwyer 's Bakery Chocolate Swiss Roll"				
Labels	`hazard`:	eggs and products thereof	`hazard_category`:	allergens
	`product`:	swiss rolls	`product_category`:	cereals and bakery products

"Smilin' Bob's Voluntarily Recalls Smilin' Bob's Smoked Fish Dip Products Because of Possible Health Risk"				
Labels	`hazard`:	listeria monocytogenes	`hazard_category`:	biological
	`product`:	fish products	`product_category`:	fish and fish products