Large language models for recipes
Final project report NLP Course, Winter 2022

Maciej Chrabaszcz
Warsaw University of Technology
maciej.chrabaszcz.stud

Aleksander Kozłowski
Warsaw University of Technology
aleksander.kozlowski.stud@pw.edu.pl

supervisor: Anna Wróblewska
Warsaw University of Technology
anna.wroblewska1@pw.edu.pl

Abstract

Large language models(LLM), such as those trained by OpenAI, can be used to classify and extract information from the text in a prompt-based manner, as shown in [3]. Given a prompt containing text data, these models can process the information and accurately identify relevant details such as dietary tags or specific ingredients. This approach allows for efficient classification and extraction of ingredients, as only a small amount of data (i.e. the prompt) needs to be processed at a time. This can be particularly useful in cases where annotating a large amount of data is infeasible or impractical and can be applied to various applications such as food labelling and recipe creation. Because of that, we will use pre-trained LLM in a prompt-based manner for the recipes domain. We aim to check how well those models can give dietary tags and extract product names without training.

1 Introduction

Using large language models (LLMs) in a prompt-based manner for the recipes domain includes the ability to accurately classify and extract relevant information from text data, such as dietary tags and specific ingredients. This approach can be applied to various applications, such as food labelling and recipe creation, and is particularly useful in situations where annotating a large amount of data is infeasible or impractical. The aim is to determine the effectiveness of using pre-trained LLMs to provide dietary tags and extract product names without additional training.

2 Related work

[3] shows that LLMs are capable of few-shot learning. In our case, we are using zero-shot learning, which was also addressed in this paper. [8] shows how to prepare prompts for LLMs, which allows for solving problems based on prompts.

3 Contribution

The approach for this project involved using pre-trained large language models (LLMs) to classify and extract relevant information from text data in the recipes domain. This was done in a prompt-based manner, with the LLMs processing individual prompts containing text data and identifying relevant details such as dietary tags or specific ingredients.

To research the effectiveness of this approach, we created prompts templates later used for tasks with [7] dataset. We then evaluated the LLMs accuracy.

In addition to evaluating the effectiveness of using pre-trained large language models (LLMs) for classification in the recipes domain, we also compared our results to a previous project in which we trained LMs specifically for this task.

4 Datasets EDA and selection

4.1 RecipeNLG

First, we analyzed [4] dataset. Unfortunately, this dataset doesn't contain any dietary tags. During our analysis, we also found out that there is a lot of noise in this data which can be seen when looking at the word cloud displayed in Figure 1.

4.2 Food.com search terms dataset

Because we wanted to classify dietary tags, we needed a dataset to allow us to do so. We found [7] dataset, which contains tags given on the food.com website to recipes. It also has ∼0.5mln records. After looking at the word cloud displayed in Figure 2, it can be seen that this dataset is well structured. To use this dataset, we also had to create a

Figure 1: RecipeNLG dataset wordcloud

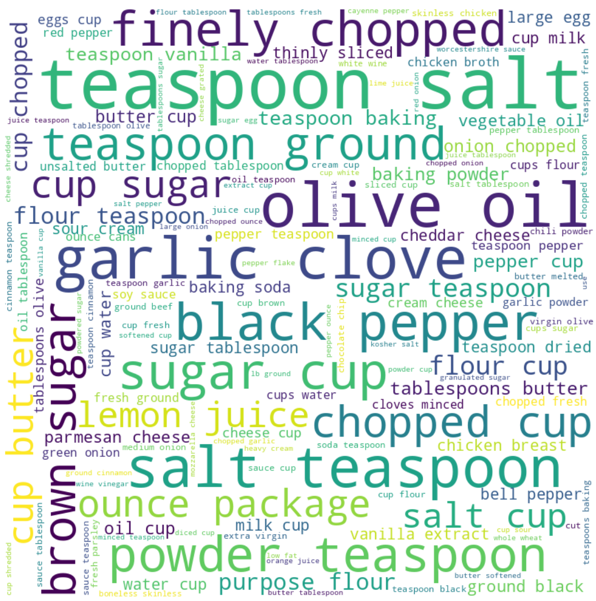

Figure 2: Food.com search terms dataset wordcloud

mapping from website tags to dietary tags of interest.

4.3 Dataset selection

Because [7] was the only dataset which allowed us to create dietary tags, we used it for our experiments.

4.4 Dataset sampling

Because of API request calls limits, we selected a sample of dataset[7]. We sampled the dataset such that there would be at least 20 of each tag. We also used only below tags for our experiments:

- vegetarian
- vegan
- sweet
- seafood
- meat
- low-sodium
- low-carb
- low-cholesterol
- low-calorie
- low-protein
- low-saturated-fat
- low-fat
- dairy
- nuts
- diabetic
- kosher

With this sampling, our final dataset has 320 examples.

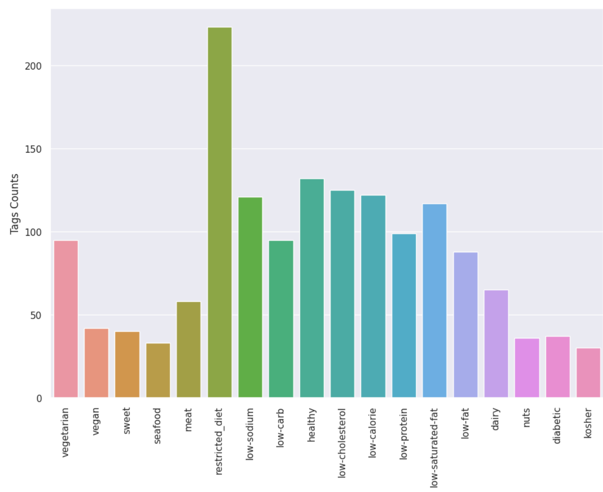

Figure 3: Dietary tags counts in sampled data

5 Experiments and Results

5.1 Dietary tags

We implemented a test when trying to classify dietary tags using LLMs by giving a model prompt asking if a given dietary tag can be assigned to recipe ingredients, which was tested automatically. Results are shown in Table 1. We can see that GPT-3 has poor knowledge related to dietary tags. One idea for improving results is to use few-shot learning instead of one-shot learning. Comparing these results to our previous project (Table 2), we can see that simple LightGBM trained on ingredient names mostly achieved better results.

5.2 Ingredients names extraction

We gave prompts to the model to extract product names from the recipe. We then compared results to ground-truth names in our dataset. When looking at examples which were tagged as wrong, we noticed that some names extracted by the model contain additional parts (ex. "shredded cheddar" instead of "cheddar"). With that in mind, we know that results in Table 3 are pessimistic.
Metrics were calculated as follows:
Ground truth ingredient in prediction: We calculated the mean amount of GT ingredients in the model prediction
Prediction ingredient in GT: We calculated the mean amount of the model predictions present in GT ingredients list.
IoU: We calculated the intersection over the union of GT and the model prediction sets.

6 Conclusions

From our experiments, it can be seen that LLMs have problems with dietary tags but work well for ingredient extraction. Especially our results in Table 1 show that, in a one-shot manner, GPT-3 has poor knowledge of dietary tags and their relation to ingredients. Surprisingly, this model could return results in JSON format given a specific prompt. In our opinion, LLMs can help resolve many problems in domains where there is a lack of data, but they lack specified knowledge in particular domains (ex. dietary tags classification). One thing we didn't try is few-shot learning; it is possible that giving such prompts could allow the model to understand this domain better. We also prepared some exciting prompts and results from LLMs in our Appendix. In those examples, we can see that these models have problems adapting recipes (It says that we should replace cabbage with something vegan-friendly), which aligns with our numerical dietary tag classification results. After our analysis, we think that LLMs can be helpful for users in the recipe domain but with humans in the loop, which could quickly find mistakes returned by those models—for example, asking these models to replace ingredients to make the recipe vegan can be helpful. We can see that this model tries to replace meat with a vegan alternative high in protein, keeping the nutritional value of the recipe similar to the base recipe. What is also worth noting is the output variation even when inputs are close to each other.

7 Work Division

Name	Work
Maciej Chrabaszcz	research and brainstorming(6h), learning how to use OpenAI API(0.5h), testing and preparing prompts for dietary tags classification(5.5h), writing scripts for dietary tags classification(1h), writing report(1.5h), testing other prompts(1.5h)
Aleksander Kozłowski	research and brainstorming(5h), learning how to use OpenAI API(1h), testing and preparing prompts for ingredients names extraction(4h), writing scripts for ingredients names extraction(1h), evaluating model responses(2h), writing report(1.5h)

References

[1] Conneau Alexis, Khandelwal Kartikay, Goyal Naman, Chaudhary Vishrav, Wenzek Guillaume, Guzmán Francisco, Grave Edouard, Ott Myle, Zettlemoyer Luke, and Stoyanov Veselin. Unsupervised Cross-lingual Representation Learning at Scale. *Cornell University - arXiv*.

Precision	Recall	Dietary Tag	F1	Accuracy	Support
vegetarian	0.57	0.73	0.64	0.76	79
vegan	0.49	0.47	0.48	0.86	46
sweet	0.22	0.97	0.36	0.62	30
seafood	0.93	1.00	0.96	0.99	25
meat	0.65	1.00	0.79	0.90	50
low-sodium	0.43	0.45	0.44	0.56	105
low-carb	0.39	0.20	0.26	0.67	81
low-cholesterol	0.60	0.48	0.53	0.66	109
low-calorie	0.61	0.37	0.46	0.66	106
low-protein	0.64	0.24	0.35	0.71	87
low-saturated-fat	0.58	0.45	0.43	0.67	76
low-fat	0.42	0.43	0.43	0.67	76
healthy	0.65	0.35	0.46	0.64	116
restricted-diet	0.00	0.00	0.00	0.29	193
dairy	0.26	0.60	0.36	0.58	53
nuts	0.30	0.42	0.35	0.82	31
diabetic	0.08	0.17	0.11	0.70	29
kosher	0.00	0.00	0.00	0.85	23

Table 1: Results for dietary tags classification using LLM.

Dietary Tag	Precision	Recall	F1	Accuracy	Support
Dairy	0.42	0.71	0.52	0.77	2172
Low-Calorie	0.46	0.77	0.59	0.78	2460
Low-Carb	0.49	0.80	0.61	0.76	2906
Low-Fat	0.41	0.79	0.54	0.83	1512
Meat	0.85	0.96	0.91	0.94	3708
Nuts	0.27	0.79	0.39	0.83	771
Plant-Based	0.43	0.84	0.58	0.78	2200
Seafood	0.90	0.94	0.93	0.99	975
Sweet	0.49	0.88	0.63	0.86	1620

Table 2: Results for LightGBM on ingredients names and TF-IDF from project 1.

Metric	Score
Ground truth ingredient in prediction	0.79
Prediction ingredient in GT	0.83
IoU	0.72

Table 3: Results for ingredients names extraction

[2] Wróblewska Ania, Kaliska Agnieszka, Pawłowski Maciej, Wiśniewski Dawid, Sosnowski Witold, and Ławrynowicz Agnieszka. Tasteset – Recipe Dataset and Food Entities Recognition Benchmark. *Cornell University - arXiv*.

[3] Tom B., Brown, Mann Benjamin, Ryder Nick, Subbiah Melanie, Kaplan Jared, Dhariwal Prafulla, Neelakantan Arvind, Shyam Pranav, Sastry Girish, Askell Amanda, Agarwal Sandhini, Herbert-Voss Ariel, Krueger Gretchen, Henighan Tom, Child Rewon, Ramesh Aditya, Daniel M., Ziegler, Wu Jeffrey, Winter Clemens, Hesse Christopher, Chen Mark, Sigler Eric, Litwin Mateusz, Gray Scott, Chess Benjamin, Clark Jack, Berner Christopher, McCandlish Sam, Radford Alec, Sutskever Ilya, and Amodei Dario. Language Models are Few-Shot Learners. *Cornell University - arXiv*.

[4] Michał Bień, Michał Gilski, Martyna Maciejewska, Wojciech Taisner, Dawid Wisniewski, and Agnieszka Lawrynowicz. RecipeNLG: A cooking recipes dataset for semi-structured text generation. In *Proceedings of the 13th International Conference*

Deliverable name	Deliverable justification
Report	Report can be found on this GitHub at file ‘NLP_project2_final_report.pdf‘. We also provide a link to our overleaf project at ‘final_report_overleaf_link.txt‘
Clean code	Our code is structured and contains comments explaining how it works. Most important files are explained in GitHub README
Reproducibility	In GitHub README, we explain how to reproduce our results step by step. Following those steps is sufficient to get exact results as we did
Contributions	Our project’s contribution tests how well LLMs work in the recipe domain. With our tests we were able to see when these models fail to work and when they thrive and can be helpful
Presentation	Our final presentation is available on GitHub in the file ”NLP_project2_final_presentation.pdf”.

Table 4: Table showing deliverables

on Natural Language Generation, pages 22–28, Dublin, Ireland, December 2020. Association for Computational Linguistics.

[5] Larissa Britto, Luciano Pacífico, Emilia Oliveira, and Teresa Ludermir. A Cooking Recipe Multi-Label Classification Approach for Food Restriction Identification. *Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2020)*.

[6] Howard Jeremy and Ruder Sebastian. Universal Language Model Fine-tuning for Text Classification. *Cornell University - arXiv*.

[7] Kaggle. Food.com recipes with search terms and tags, 2021.

[8] Reynolds Laria and McDonell Kyle. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. *Cornell University - arXiv*.

[9] Shanahan Murray. Talking About Large Language Models. *Cornell University - arXiv*.

[10] Diwan Nirav, Batra Devansh, and Bagler Ganesh. A Named Entity Based Approach to Model Recipes. *Cornell University - arXiv*.

A Additional prompts

A.1 NER

We tried using LLMs for NER, but because the output was unpredictable in our setting, we could not create any tests for models. For further research, one could try to use LLMs for this task in a few-shot learning manner. Below you can see examples of input and model result

Input: Do Named Entity Recognition on below text and return result in JSON format
text:
1/3 cup toasted pumpkin seeds. 4 ounces softened reduced-fat cream cheese or 4 ounces cream cheese. 4 ounces goat cheese (I used La Buchette with Fine Herbs). 1 (16 ounce) jar salsa (Frontera Jalapeno Cilantro recommended). chopped fresh cilantro
result:

Output:
```
{
”entities”: [
{
”text”: ”1/3 cup”,
”type”: ”Quantity”
},
{
”text”: ”toasted pumpkin seeds”,
”type”: ”Food”
},
{
”text”: ”4 ounces”,
”type”: ”Quantity”
},
{
”text”: ”reduced-fat cream cheese”,
”type”: ”Food”
},
{
”text”: ”4 ounces”,
”type”: ”Quantity”
},
{
”text”: ”cream cheese”,
```

”type”: ”Food”
},
{
”text”: ”4 ounces”,
”type”: ”Quantity”
},
{
”text”: ”goat cheese”,
”project’sFood”
},
{
”text”: ”La Buchette with Fine Herbs”,
”type”: ”Food”
},
{
”text”: ”1 (16 ounce)”,
”type”: ”Quantity”
},
{
”text”: ”jar salsa”,
”type”: ”Food”
},
{
”text”: ”Frontera Jalapeno Cilantro”,
”type”: ”Food”
},
{
”text”: ”chopped fresh cilantro”,
”type”: ”Food”
}
]
}

A.1.1 Ingredients replacement

We were curious if it is possible to use LLMs for ingredients replacement to make recipe fit specified diet. Again one could try using it on bigger scale with human annotator to check how well these models work in this area. We only tested few examples which are listed below

Input: How to replace below ingredients to make it Vegan?
2 tablespoons olive oil. 1 cup onion (chopped). 2 -3 garlic cloves. 1 lb stewing beef. 1 (32 ounce) box beef broth. 1/2 cup red wine. 1/2 cup water. 1 cup barley. barley, is not gluten free. 2 bay leaves. ground black pepper. 1 -2 dash Worcestershire sauce. 1 cup cherry tomatoes (halved, or a can of chopped tomatoes). 8 ounces sliced mushrooms (crimini best). 1 cup cabbage, 1 inch dice

Output: To make this recipe vegan, replace the following ingredients:
-Olive oil: Replace with vegan-friendly oil such as coconut oil or avocado oil.
-Stewing beef: Replace with vegan-friendly protein such as tempeh, seitan, or tofu.
-Beef broth: Replace with vegan-friendly vegetable broth.
-Red wine: Replace with vegan-friendly red wine or a non-alcoholic alternative.
-Barley: Replace with a gluten-free grain such as quinoa or millet.
-Worcestershire sauce: Replace with vegan-friendly Worcestershire sauce.
-Cherry tomatoes: Replace with a can of diced tomatoes or fresh tomatoes.
-Mushrooms: Replace with vegan-friendly mushrooms such as shiitake or oyster mushrooms.
-Cabbage: Replace with vegan-friendly vegetables such as kale, spinach, or bell peppers.

Input: How to replace Stewing beef and Beef Broth to make below recipe Vegan?
2 tablespoons olive oil. 1 cup onion (chopped). 2 -3 garlic cloves. 1 lb stewing beef. 1 (32 ounce) box beef broth. 1/2 cup red wine. 1/2 cup water. 1 cup barley. barley, is not gluten free. 2 bay leaves. ground black pepper. 1 -2 dash Worcestershire sauce. 1 cup cherry tomatoes (halved, or a can of chopped tomatoes). 8 ounces sliced mushrooms (crimini best). 1 cup cabbage, 1 inch dice

Output: To make the recipe vegan, replace the stewing beef with a vegan alternative such as seitan, tempeh, or tofu. Replace the beef broth with a vegan broth such as vegetable broth or mushroom broth. Replace the Worcestershire sauce with a vegan alternative such as tamari or soy sauce. Replace the red wine with a vegan alternative such as apple cider vinegar or balsamic vinegar.

Input: How to replace below ingredients to make it vegetarian?
2 tablespoons olive oil. 1 cup onion (chopped). 2 -3 garlic cloves. 1 lb stewing beef. 1 (32 ounce) box beef broth. 1/2 cup red wine. 1/2 cup water. 1 cup barley. barley, is not gluten free. 2 bay leaves. ground black pepper. 1 -2 dash Worcestershire sauce. 1 cup cherry tomatoes (halved, or a can of chopped

tomatoes). 8 ounces sliced mushrooms (crimini best). 1 cup cabbage, 1 inch dice

Output: To make this recipe vegetarian, you can replace the stewing beef with a plant-based protein such as seitan, tempeh, or tofu. You can also replace the beef broth with vegetable broth. The Worcestershire sauce can be replaced with a vegan Worcestershire sauce. The barley can be replaced with quinoa or brown rice. The mushrooms can be replaced with other vegetables such as bell peppers, zucchini, or eggplant. Finally, the cabbage can be replaced with kale or spinach.

Input: What are some ways to adapt this recipe to be vegan?
2 tablespoons olive oil. 1 cup onion (chopped). 2 -3 garlic cloves. 1 lb stewing beef. 1 (32 ounce) box beef broth. 1/2 cup red wine. 1/2 cup water. 1 cup barley. barley, is not gluten free. 2 bay leaves. ground black pepper. 1 -2 dash Worcestershire sauce. 1 cup cherry tomatoes (halved, or a can of chopped tomatoes). 8 ounces sliced mushrooms (crimini best). 1 cup cabbage, 1 inch dice

Output: To make this recipe vegan, you could substitute the beef with a vegan alternative such as seitan, tempeh, or tofu. You could also use vegetable broth instead of beef broth. Additionally, you could omit the Worcestershire sauce and replace it with a vegan alternative such as tamari or soy sauce. Finally, you could replace the olive oil with a vegan alternative such as coconut oil.

Input: What are some ways to adapt this recipe to be vegan?
2 tablespoons olive oil. 1 cup onion (chopped). 2 -3 garlic cloves. 1 lb stewing beef. 1 (32 ounce) box beef broth. 1/2 cup red wine. 1/2 cup water. 1 cup barley. barley, is not gluten free. 2 bay leaves. ground black pepper. 1 -2 dash Worcestershire sauce. 1 cup cherry tomatoes (halved, or a can of chopped tomatoes). 8 ounces sliced mushrooms (crimini best). 1 cup cabbage, 1 inch dice

Output: 1. Replace the beef broth with vegetable broth.
2. Use leaner cuts of beef, such as sirloin or flank steak.
3. Use brown rice instead of barley.
4. Use low-sodium Worcestershire sauce.
5. Add more vegetables, such as carrots, celery, and bell peppers.
6. Use olive oil or avocado oil instead of butter.
7. Use fresh herbs instead of dried herbs.
8. Use low-fat or non-fat dairy products, such as Greek yogurt or cottage cheese.
9. Use low-sodium canned tomatoes.
10. Add legumes, such as chickpeas or lentils, for added protein and fiber.

Even from these few examples, we can see that this model doesn't fully understand the recipe domain. For example, it says that we should replace olive oil with a vegan-friendly option, but it's already vegan.

A.1.2 Nutritional breakdown

We also tried to ask GPT-3 to do a nutritional breakdown of the recipe. You can see an example below

Input: Can you give me a nutritional breakdown for this recipe?
16 ounces of fresh Chinese egg noodles (Lo mein noodles)
3 tablespoons vegetable oil
6 garlic cloves, minced
8 ounces pork tenderloin, thinly sliced
2 teaspoons rice wine vinegar
6 -8 cups water
3 tablespoons soy sauce
2 tablespoons dashida, powder (Beef flavored soup stock)
2 teaspoons ginger, grated
2 cups cabbage kimchi, chopped in big pieces
4 ounces shiitake mushrooms, thinly sliced
7 ounces firm tofu, cut into 1-inch squares
4 baby bok choy, quartered
2 scallions, thinly sliced
2 teaspoons dark sesame oil
salt and pepper (Add a little salt and pepper at a time and taste as you go. The kimchi and soy sauce may be enough s)
sriracha sauce (Optional but highly recommended.)

Output: Nutritional breakdown per serving (4 servings):
Calories: 441
Fat: 22.3g

Carbohydrates: 39.3g
Protein: 22.3g
Sodium: 1290mg
Fiber: 5.2g
Sugar: 5.2g

Comparing this result with the real nutritional breakdown given to this recipe (Figure 4) shows that model completely fails at this task

NUTRITION INFO

Serving Size: 1 (683) g
Servings Per Recipe: 4

AMT. PER SERVING	% DAILY VALUE
Calories: 685.6	
Calories from Fat 197 g	29 %
Total Fat 21.9 g	33 %
Saturated Fat 4.1 g	20 %
Cholesterol 132.6 mg	44 %
Sodium 854.9 mg	35 %
Total Carbohydrate 88.3 g	29 %
Dietary Fiber 5.8 g	23 %
Sugars 4.1 g	16 %
Protein 35.1 g	70 %

Figure 4: Nutritional breakdown of the recipe given to model

B Confusion matrices for dietary tags classification

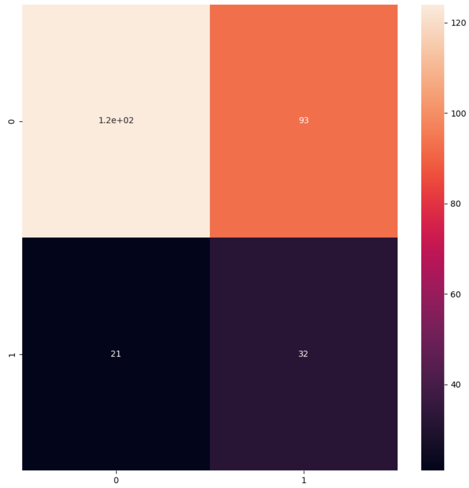

Figure 5: Dairy tag confusion matrix. X-axis is true label.

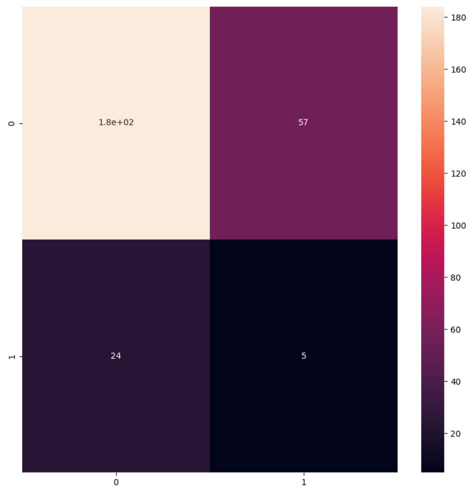

Figure 6: Diabetic tag confusion matrix. X-axis is true label.

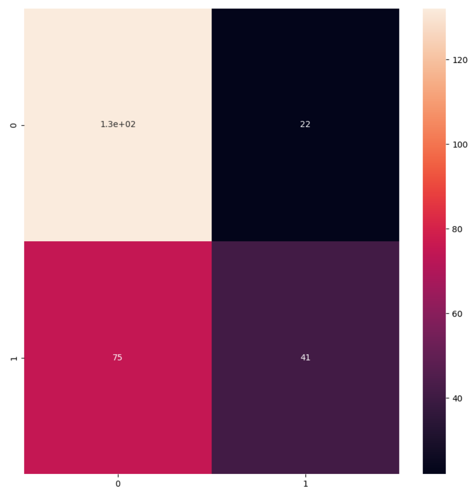

Figure 7: Healthy tag confusion matrix. X-axis is true label.

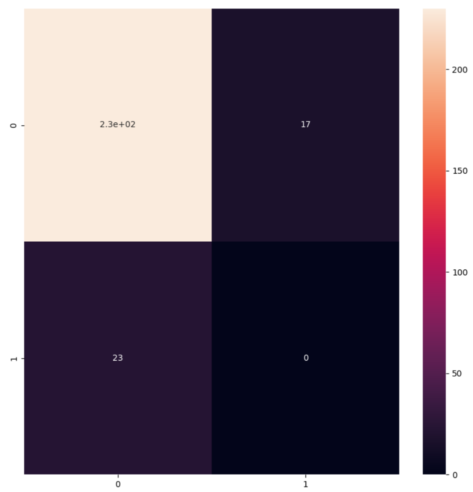

Figure 8: Kosher tag confusion matrix. X-axis is true label.

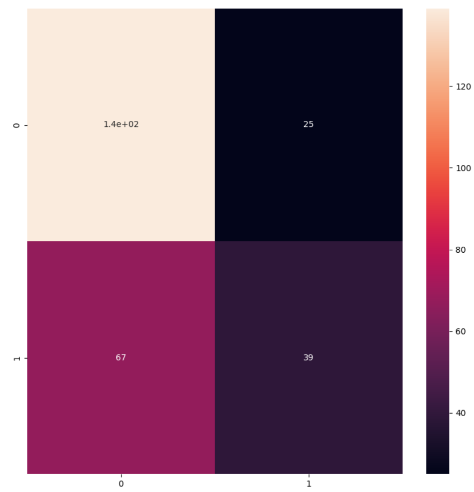

Figure 9: Low-calorie tag confusion matrix. X-axis is true label.

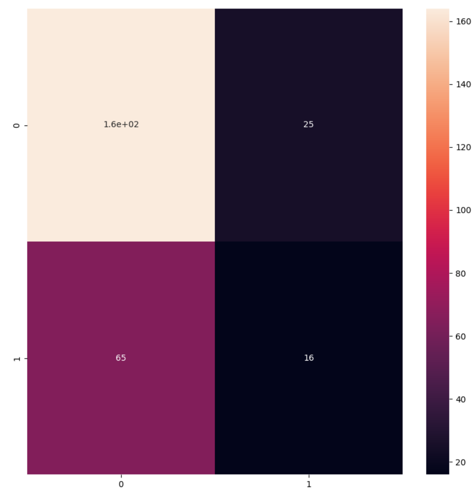

Figure 10: Low-carb tag confusion matrix. X-axis is true label.

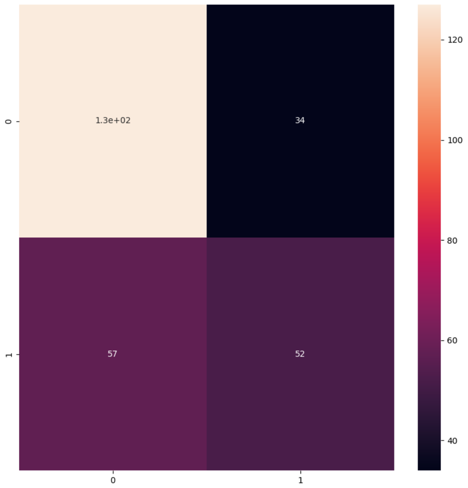

Figure 11: Low-cholesterol tag confusion matrix. X-axis is true label.

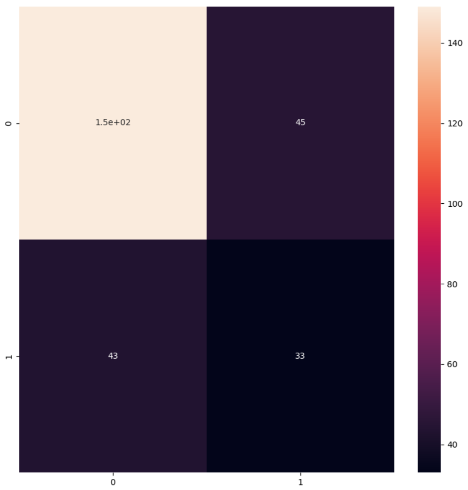

Figure 12: Low-fat tag confusion matrix. X-axis is true label.

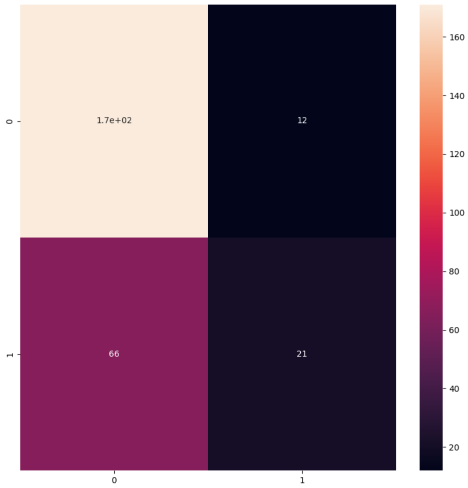

Figure 13: Low-protein tag confusion matrix. X-axis is true label.

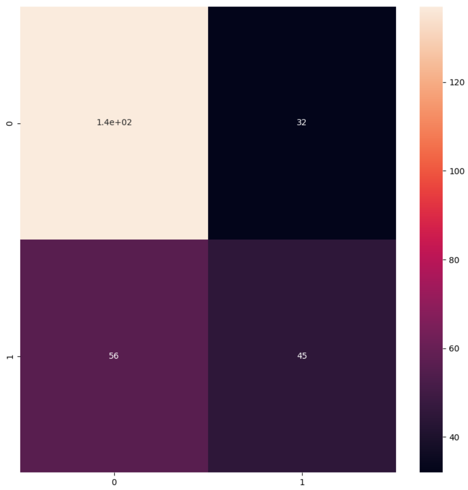

Figure 14: Low-saturated-fat tag confusion matrix. X-axis is true label.

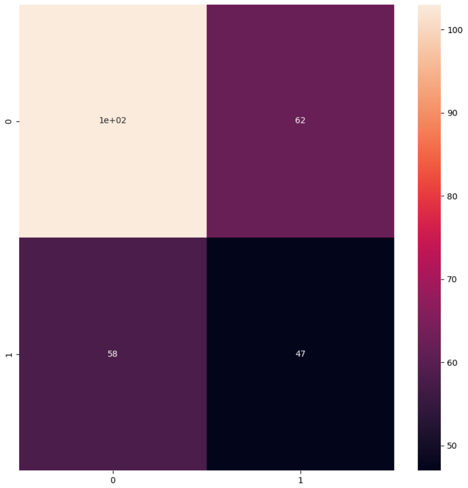

Figure 15: Low-sodium tag confusion matrix. X-axis is true label.

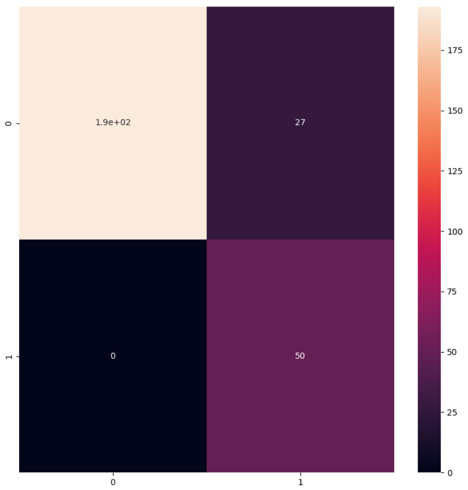

Figure 16: Meat tag confusion matrix. X-axis is true label.

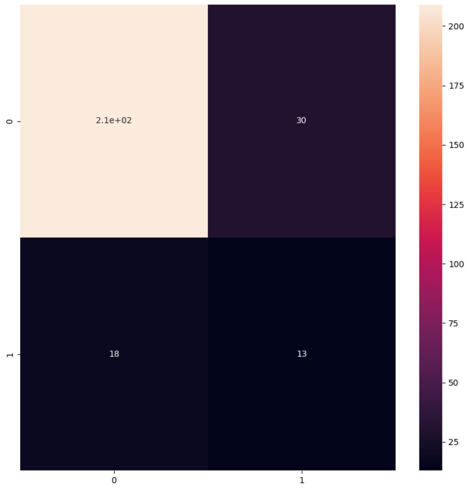

Figure 17: Nuts tag confusion matrix. X-axis is true label.

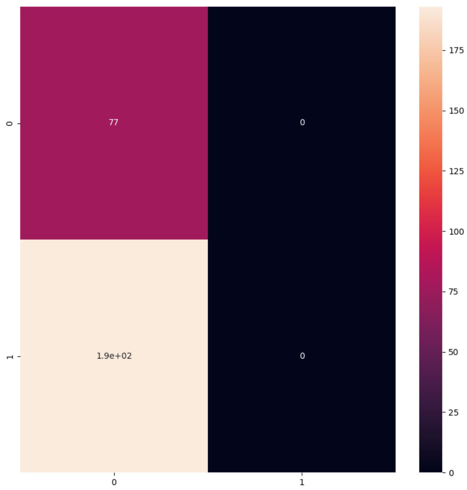

Figure 18: Restricted-diet tag confusion matrix. X-axis is true label.

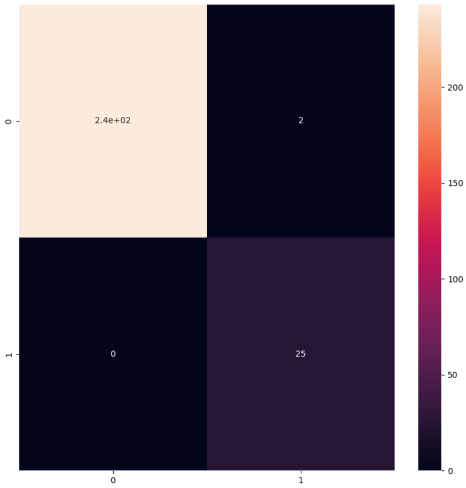

Figure 19: Seafood tag confusion matrix. X-axis is true label.

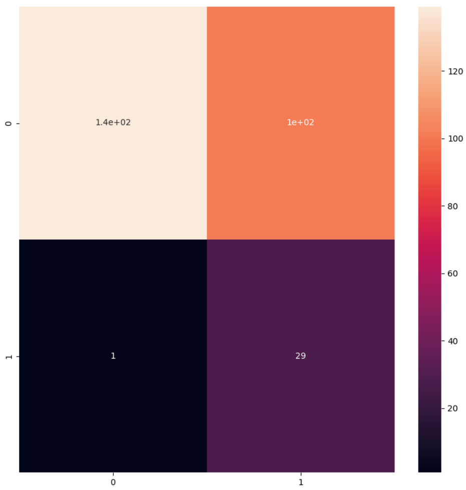

Figure 20: Sweet tag confusion matrix. X-axis is true label.

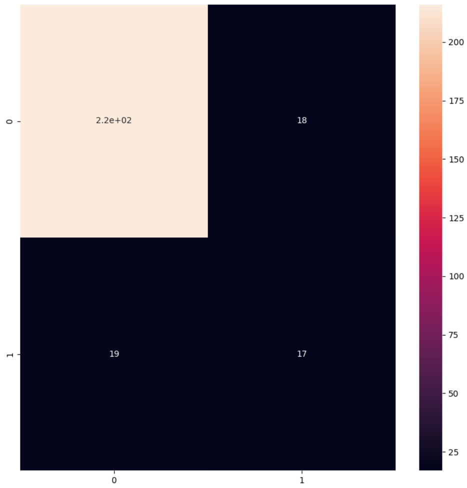

Figure 21: Vegan tag confusion matrix. X-axis is true label.

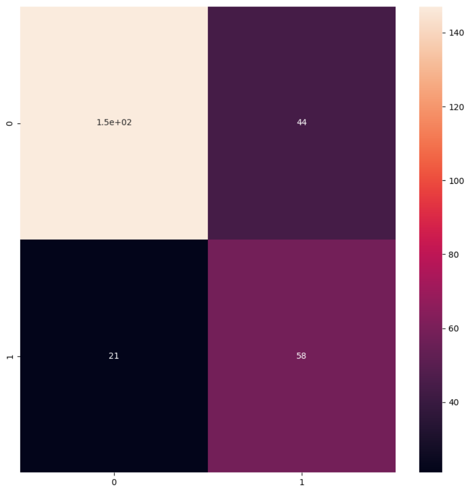

Figure 22: Vegetarian tag confusion matrix. X-axis is true label.