AllerMatch - Final Report

CSCE 470-500

Team Members

Aashish Ananthanarayan
Matthew Kanarr
Dina Mayett
Priya Patel

Table of Contents

1. Introduction

Due to the COVID-19 pandemic a lot of people prefer to cook at home instead of going out to avoid catching a virus. But people don't have time to browse through thousands of recipes online trying to find the best recipe that is safe for them to eat as it avoids their food allergies. AllerMatch is a tool that aims to solve the problem of finding recipes that match your allergy information, keeping in mind any specific ingredients you may want. Currently, there's a couple different apps that can scan products to find out if you may be allergic to an ingredient in the product. However, we didn't see an existing tool that did the same for recipes. We wanted users to be able to search for recipes that avoided their allergens and recommend some recipes that they can enjoy.

1.1 Problem Statement

32 million people are affected by food-related allergies in the U.S alone. 1 in 12 children are diagnosed with food allergies in the U.S. and over 50% of adults report developing at least one food allergy after the age of 18. Users often end up searching for recipes only to find they're allergic to one of the ingredients, or they don't have the time to browse through countless recipes just to look for one that aligns with their allergens. This is where our tool, AllerMatch, comes into the picture. Our target audience is anyone who likes to cook and wants homemade meals but has to keep track of allergy information while searching for recipes, or doesn't have the time to browse. We wanted to provide our target audience with an easy and efficient way to search for the recipes they need, while keeping their allergen information in mind.

1.2 Our Solution

AllerMatch is a recipe tool that takes into consideration your allergies and food preferences, and generates the top six recipes matching your inputs. It's an easy to use application that only requires you to input your allergy information, any food preferences (for example, if you have an ingredient that’s lying around the fridge) and the recipe name you want to look for. Then the application will produce the best six matching recipes, and you can select one to enjoy!

2. Data Collection

For our data, we first thought of crawling through recipe webpages like allrecipes.com in order to train the model. However, we decided to take a different approach and use the edamam API in order to collect our data according to a query parsed and then run the algorithms on them. We decided to dynamically work on the data and rather than having loads of data, we can simply use the API to get data according to our needs.

The data is received in the form of a json file which can then be easily parsed and used according to our needs. We utilized several fields of the data in our algorithms. How the edamam API works is pretty simple. The url simply contains the name of the recipe parsed, for which hundreds of recipes are returned with their ingredients, nutritional information, contents, regulations and so on.

3. Approach

Our goal was to build a simple web application that could take a user’s allergy requirements and then recommend a small amount of recipes that they could use. In order to do this, we decided to use two algorithms, cosine similarity and Okapi BM25 that we learned in class. For BM25 we used Python’s open source packet and made modifications to it as well in order to see how well it performed and tested the algorithm accordingly.

Our application takes in three inputs from the user. The first is the dish they want to cook, the second is their allergies and the third is any particular ingredients they'd like to have in the dish. The choice to select their ingredients is optional, but this option is crucial in selecting which algorithm to use for deciding the recommendations.

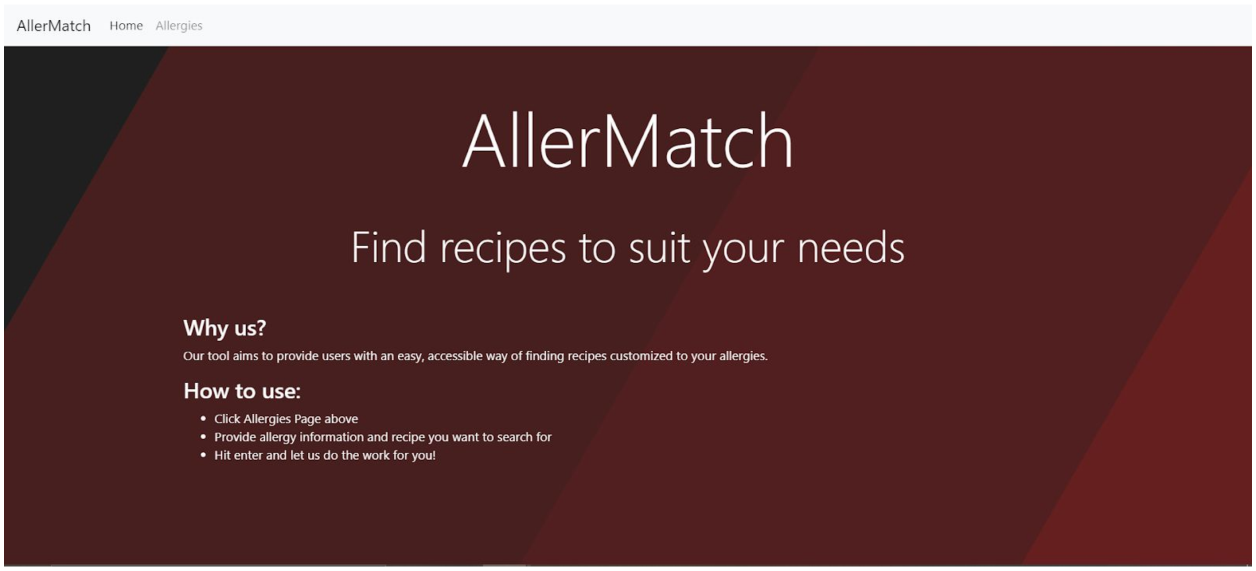
If the user does not select any ingredients, the application will use cosine similarity between the ingredients in the API to select six different recipes that the user can use. First, the API pulls all the recipes from the given recipe query. Then, a check is made to remove all the recipes that contain the allergens inputted from the data. Once this is done, we are left with a number of recipes to choose from. A recipe from these is selected and then, cosine similarity is performed with that recipe weighted the most so that five other recipes extremely similar to that one are chosen. These recipes are then displayed as the output. Since our application is mostly focused on providing good choices of recipes based on ingredients, we performed cosine similarity on the ingredients of the recipes by generating ingredient vectors.

The second option was if users decided to input their ingredient choices into the application. For this case, we decided to use the BM25 algorithm. The user query for ingredients was taken and then, the documents were ranked according to the algorithm and the six most relevant recipes were displayed.

4. Implementation

Our team wanted to create and deploy a scalable application as soon as possible. To achieve this, we use the webapp framework known as Flask to handle routing and passing of our data obtained from the Edamam API. Having a web app in mind we wanted our application to be accessible from various devices. To achieve this we used the CSS framework known as Bootstrap. This framework helped us design a responsive web application that can be accessed while on your computer or on the go on your mobile device. For our backend we used python as Flask is a Python webapp framework. Our backend handled the API request and processed our corpus to obtain a user's recommended recipes based on their provided allergy and recipe information. To host our application we used the service known as pythonanywhere. Lastly, we curated our codebase on GitHub using Git for source control.

4.1 User Interface



AllerMatch

Find recipes to suit your needs

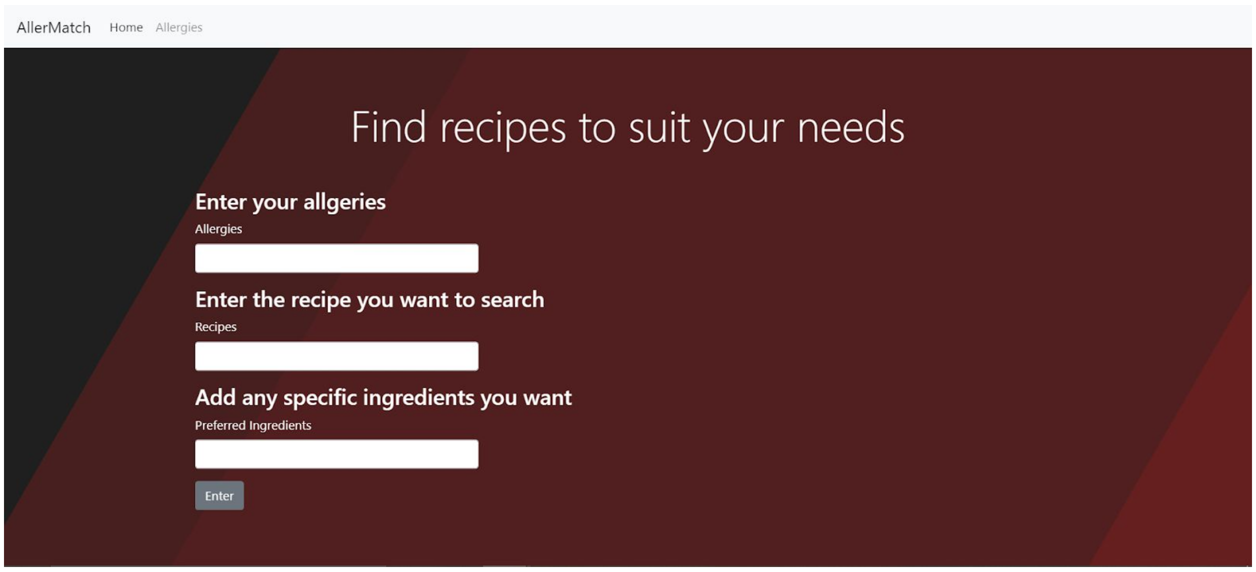
Why us?

Our tool aims to provide users with an easy, accessible way of finding recipes customized to your allergies.

How to use:

- Click Allergies Page above
- Provide allergy information and recipe you want to search for
- Hit enter and let us do the work for you!

Fig 1. Landing page



Find recipes to suit your needs

Enter your allgeries

Allergies

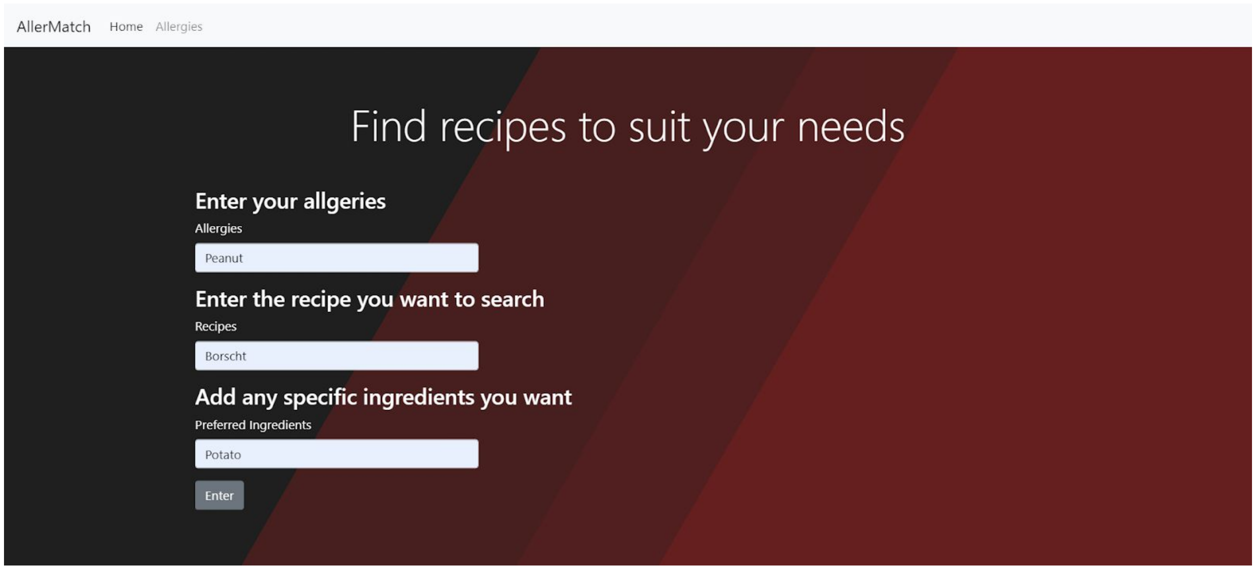
Enter the recipe you want to search

Recipes

Add any specific ingredients you want

Preferred Ingredients

Fig 2. Allergies page with blank form


AllerMatch Home Allergies

Find recipes to suit your needs

Enter your allgeries

Allergies: Peanut

Enter the recipe you want to search

Recipes: Borscht

Add any specific ingredients you want

Preferred Ingredients: Potato

Enter

Fig 3. Allergies page with sample query

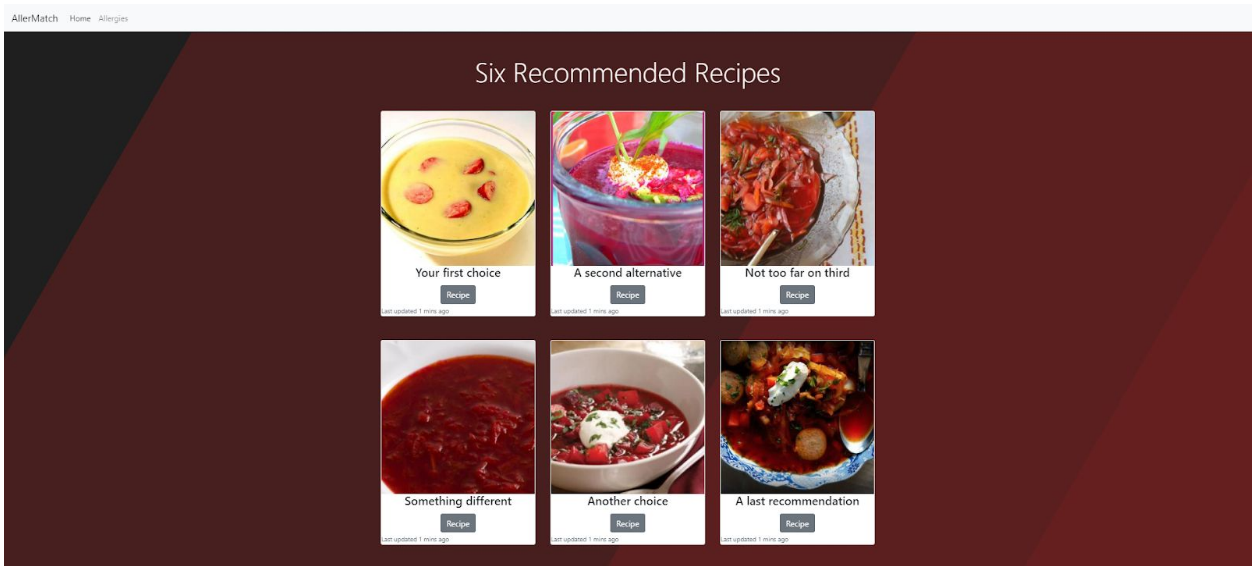


Fig 4. Results page

4.2 Demo Link

https://www.youtube.com/watch?v=xTH3KWF8Eus&feature=emb_logo

5. Evaluation

To evaluate our Cosine Similarity and Okapi BM25, we used nDCG scoring. In order to perform this scoring, we use the recipe the user is searching for as the base for both BM25 and Cosine Similarity.

If the user did not provide specific ingredients they want in their recipe, we used Cosine Similarity to return six different recipes that are similar to our base. The recipes returned in this scenario are in no particular order. When the user provides ingredient information, the Okapi BM25 Algorithm is used to provide the top 6 recipes listed in descending order. In this scenario we compare the recipes that contain our ingredient to the base. We hit ~0.74 NDCG value on Cosine similarity and BM25 hovered a little lower.

5.1 Drawbacks

Unfortunately, the Edamam API version we used was free and hence It provided us a limited number of API calls per month and per minute. It made the testing and development process more complicated and difficult.

Another problem that we encountered during the testing was that the BM25 used partitioned words in the recipe documents to match words in the query instead of using the exact word match.

Ex. -query: “potatoes”
-recipe: “pot”

6. Conclusion

Overall, our tool successfully provides the top six results for the user’s allergy information and recipe keeping any food preferences in mind. We are very happy with the progress we made during this semester and we got a chance to implement some of the ranking algorithms taught in class in real world scenarios. With the application we developed, we successfully implemented Cosine Similarity and Okapi BM25 in providing relevant results to our user based on their query.