The SpiceRack

EECS Senior Design Expo

Developers: Ben Hollar, Erin Graska, Stephanie Tam

Advisor: Dr. Chia Han

GOALS

Project Overview

Finding, collecting, and cooking recipes can be a daunting task. There are numerous hurdles for people to effectively manage their recipes.

The SpiceRack is a **web-application** dedicated to parsing various sources of recipes and collecting them for a user in a **digital recipe book.**

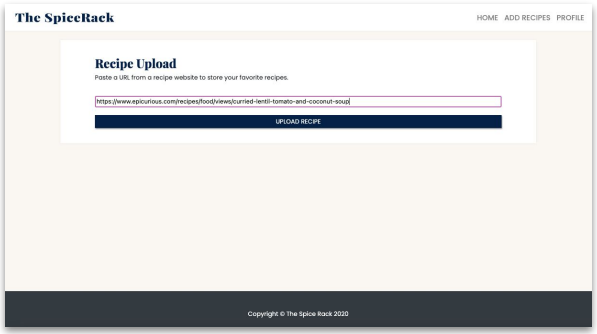
Recipes will be interpreted in one of two ways: For online recipes (i.e., the recipe source is a URL), we aim to use **machine learning** and **information retrieval** techniques to strip recipe information from the webpage.

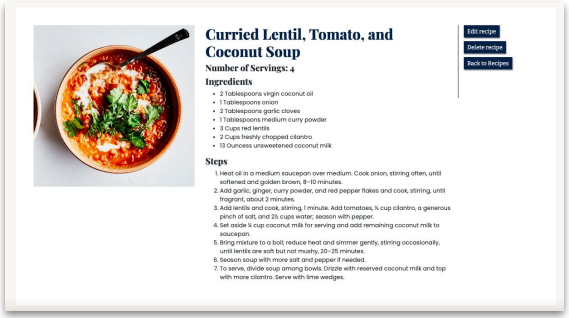
Team Goals:

- Gain experience in machine learning/information retrieval techniques, creating a database that incorporates REST API, and creating a user-friendly webpage that showcases our site's main features.

INTELLECTUAL MERITS

1. Devised machine learning models for extracting recipe content from webpages
   a. Leverages existing ML frameworks
      i. Dragnet: web page content extraction
      ii. BERT: state-of-the-art Natural Language Processing for text classification
   b. Built on a novel database of 360+ publicly available recipes
2. Created a web application to leverage our ML models and make collecting recipes as simple and convenient as possible!





BROADER IMPACTS

Since the rise of the pandemic, more individuals are cooking more frequently at home, and even coming up with their own unique recipes. We hope that users can use our web application to document their cooking journey.

How does The SpiceRack differ from a typical recipe website?

The SpiceRack allows for quicker recipe retrieval:

An ‘all in one’ recipe storage app, minus the hassle of scrolling through ads or useless text.

How do we make the web application more personalized to the user?

The web application is customizable to whatever recipe the user has at hand (i.e adding recipes of choice)

The SpiceRack allows user to store written recipes in an online easy to read format, or upload recipes based on their preferences. Users who possess traditional hand-written recipes will have a way to store the recipes that can be passed down later on.

DESIGN SPECIFICATIONS OVERVIEW

WEB-APPLICATION

This is the user-interface and backing server/database for users to collect and view their saved recipes. It was built in Python 3 using the Django web-development framework.

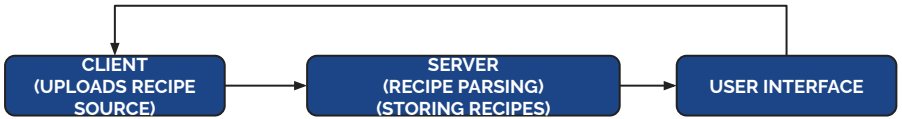
MACHINE LEARNING

All of the machine learning development, including training and validation data. This section itself was split into two parts:

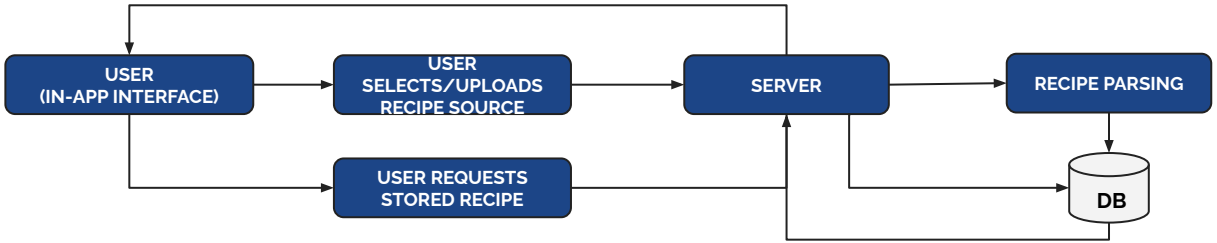
1. Content extraction, to retrieve recipe text from webpages
2. Text classification, to create a uniform structure from retrieved recipe text

DESIGN DIAGRAM

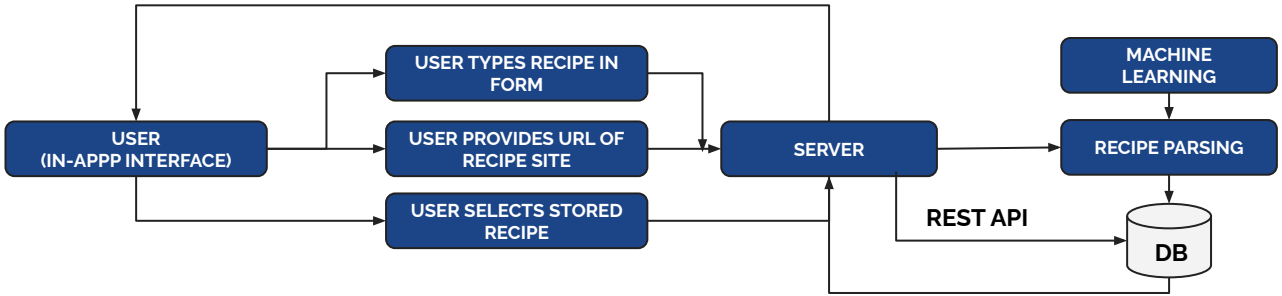
D0



D1



D2



TECHNOLOGIES USED

DJANGO

Managing Server/Client relationships

BOOTSTRAP 4

UI Design and styling with CSS

MACHINE LEARNING + WEBSITE SCRAPING

DRAGNET

Web page content extraction

See: Dragnet

FAST-BERT

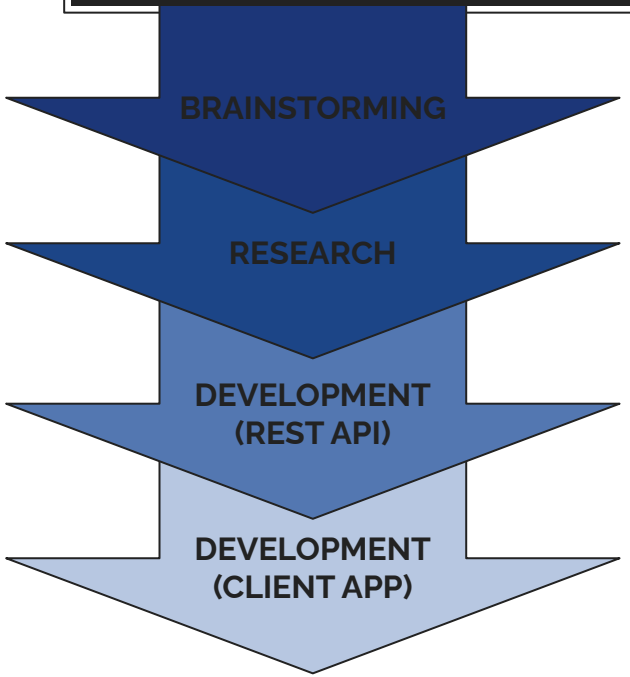
Framework for simple setup of BERT (state-of-the-art NLP)

See: fast-bert

PYTHON

Utilizing Python ML Libraries, more specifically scikit

MILESTONES



(1) Brainstorming: What features does the application include? How will they work? (2020-12-06 to 2021-01-10)

Task	Start Date	Completion Date
Investigate methods of parsing websites for specific data	2020-12-06	2020-12-20
Investigate optical character recognition (OCR) techniques and implementations.	2020-12-06	2020-12-20
Design `Recipe` model -- what information could a recipe include?	2020-12-13	2020-12-20
Identify programming language, frameworks (if any), etc. to implement server.	2020-12-21	2021-01-10
Develop relational database to store `Recipe` objects.	2020-12-21	2021-01-10

(2) Research: Develop parsing, OCR strategies specific to our application. (2021-01-10 to 2021-02-07)

Task	Start Date	Completion Date
Research and prototype parsing strategies identified in language of choice.	2021-01-10	2021-02-07
Research and prototype OCR parsing functionality investigated previously.	2021-01-10	2021-02-07

(3) Development: Data Transfer, Authentication (2021-02-07 to 2021-03-21)

Task	Start Date	Completion Date
Develop REST API to pass data between server and client application(s).	2021-01-23	2021-01-25
Develop user-management functionality	2021-02-01	2021-02-02

(4) Development: Client Application (2020-12-06 to 2021-03-21)

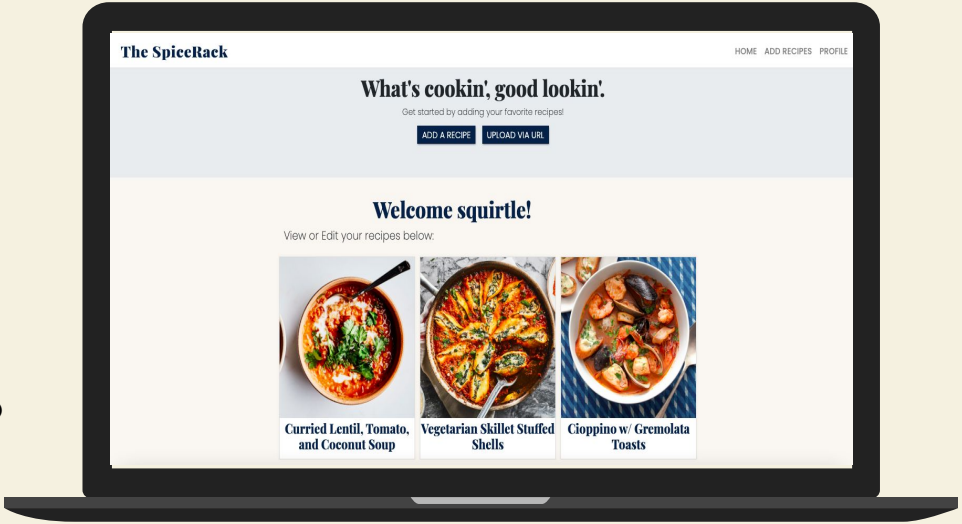
Task	Start Date	Completion Date
Define web-development framework to create the application.	2020-12-13	2020-12-20
Define CSS and other design principles	2020-12-13	2020-12-20
Create "user-management" page(s) for signing up, logging in, etc.	2021-01-10	2021-01-24
Create "home" webpage.	2021-01-25	2021-01-30
Create "recipe viewer" webpage.	2021-01-25	2021-01-30
Create form to upload recipe source to be parsed and stored.	2021-02-14	2021-03-07
Create page(s) to manually create, edit, and delete recipes	2021-01-25	2021-01-30

CHALLENGES AND SOLUTIONS

1. Displaying data from user created by a form
    a. We used a Django backend that took care of any user entry. This includes (but is not limited to): adding a recipe, editing a recipe, creating a user
2. Motivation to meet milestones
    a. Setting up meetings twice a month allowed us to talk about any updates, and also assigning new tasks that follow our milestones list.
3. Parsing recipes completely (i.e. retrieving only necessary information from websites with high precision)
    a. A training model we currently have is being modified to successfully extract recipe information from any webpage.
    b. We opted to focus on high precision, since it will return only the information we need.

RESULTS

- Use Machine learning to parse recipe parts from web pages
- Created functioning web server with Django that has the following:
  - User Account creation/management
  - Recipe creation/management
  - Database for User Accounts and Recipes to be stored in
- Implemented UI Design principles to create a functional UI for users



FUTURE GOALS

- Optical Character Recognition
- Expanding our ML database

Thank You!

ANY QUESTIONS?

You can visit our FAQ in our Github Repository, or contact us:

Ben Hollar: hollarbl@mail.uc.edu

Erin Graska: graskaen@mail.uc.edu

Stephanie Tam: tamse@mail.uc.edu