

CS 145 Milestone 3 - ToMEto

Jonathan Joo, Matthew Jin, Boyu (Charlie) Tong, Albert Ge

May 9, 2016

1 Goals

This week, we wanted to accomplish

1. Implement (more) algorithms
2. Improve data quality
3. Integration of frontend with backend

2 Progress

Our project now has all the basic components completed and integrated together. We have completed all the goals for these two weeks, and now have a working prototype for our basic use case!

2.1 Implement (more) algorithms

We have implemented/tested several algorithms at this point, and can easily switch between them when starting up the website. We have not yet determined a “best” algorithm, and are still trying out more ideas:

- (DONE) Naive, degree-based
- (DONE) Pairwise PMI
- Generalized PMI
- PMI with minimax or other constraints

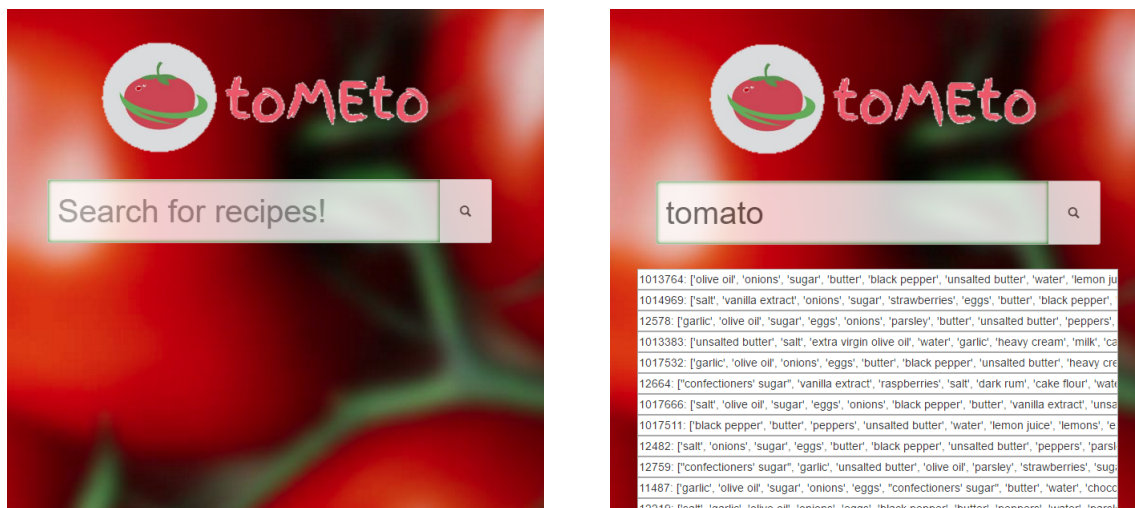
2.2 Improve data quality

We implemented an ingredient preprocessor that maps “low quality” or “noisy” ingredients to better ingredients. The goal is to limit the set of ingredients and get a better representation of the actual components for each recipe. For example, some mappings are:

- medium-large shrimps (about 14 to 16) → shrimps
- slices freshly toasted french bread → french bread
- small potatoes, cooked, peeled and sliced → potatoes

2.3 Integration of frontend with backend

The website is now fully functional for our basic use case. There is a tab where the user can input search queries for recipes. The queries are forwarded to the backend, where it is processed into relevant recipe results. For each recipe returned, the frontend can display the recommended ingredients to add.



We currently forward the search query to the actual NYT website and use their search results. We thought this was the simplest and best way to handle the queries without having to develop our own search engine.

2.4 Design improvements

We also refactored our codebase in order to make the entire flow of the program much easier to picture and understand. There are now distinct components for everything that we do, modularized in a logical way which allows for easy modification, testing, and integration:

- Data scraper + parser: `scraper.py`, `parser.py`
- Ingredient preprocessor: `mapper.py`
- Ingredient recommender: `analyzer_*.py`
- Frontend: `app.py`

3 Contributions

Jonathan Joo: Worked on integrating backend with frontend via search.

Matthew Jin: Worked on backend of search and mapper, and researched Amazon Mechanical Turk.

Charlie Tong: Worked on `mapper.py`, code refactoring, and frontend/backend integration.

Albert Ge: Worked on more algorithms (generalized PMI, PMI with constraints).

4 Adjustments to plan

We have improved our results since the last milestone by improving data quality and minor adjustments to our algorithms. On the backend, we will probably try more algorithms and ideas. This includes expanding on our current implementations (naive, PMI) by trying different variants. We will also do some more research and see if there are other techniques that we can utilize.

On the frontend, the GUI and interface are functional at the basic level. We will continue to improve the integration and include more features. Right now the data is presented as a list, which is clearly not optimal for the user. We will make the search results more presentable and have an actual landing page for each recipe. This will be in line with our fourth milestone goal to “Complete GUI front-end”.