Cuisine Classification by Recipe

Brian Cheang (*bmc78*), David Jin (*dj256*), Alvin Qu (*aq38*)

Keywords: machine learning, text classification, multi-class

Proposal

Our goal is to build a multi-class cuisine classification model for cooking recipes. The model will take a recipe as input and output cuisine classification (ex: "Greek", "Korean" …). The training data will consist of pre-labeled datasets of recipes and corresponding cuisine types. To increase the amount of training data, we plan to scrape online recipe databases to extract ingredients and cuisine types as well. Initially, we plan to use a list of ingredients (array of strings) as the input to the model but will explore incorporating other features (if available) to improve the model's predictive ability. We will implement and evaluate the model using the various Python machine learning resources and libraries (scikit-learn, tensorflow, keras…)

To allow users to access the model, we plan to implement an interactive demo. At the most basic level, this will allow users to enter ingredients and receive instant cuisine class prediction from the trained model. We are considering implementing other features as well (recipe recommendation, dynamic cuisine percentage match, ...) if time permits.

Evaluation

In order to have an effective final system that optimizes prediction accuracy, we have to develop an objective evaluation method that can verify whether or not our model is successful, and to what extent, throughout the training process.

To achieve this, we will first split the dataset we produce from combining Kaggle data and web-scraped data into 3 subsets: a training set, a validation set, and a test set. To ensure objectivity, we plan to split the dataset at random during pre-processing. Following conventional ML practices, we intend to split the entire dataset with a proportion of 80%, 10%, and 10%, respectively. As for the inclusion of the validation set, we wish to fine-tune the parameters of our model and avoid potential over-fitting of the model on the training set prior to evaluation using the testing set.

Beyond producing a testing set, we need to formulate a metric that precisely evaluates how successful our system is, which entails the construction of a baseline model that we can compare our model to. By design, one such plausible baseline model will analyze the proportions of labels in the testing set (for example, 70% *Chinese* and 30% *Greek*) and output predictions by assigning labels to the testing set at random using the same proportions. In comparing the two models, we want to ensure that our model outperforms the baseline model in terms of total prediction accuracy across the entire testing set. Additionally, we also wish to more specifically focus on setting a baseline for the accuracy for each of the separate classes by producing accuracy scores within each ethnicity label for the recipes (for example, what percentage of the Japanese labels were we successful in predicting?). This can guarantee that our model is capable of achieving a certain accuracy across all classes and avoids the possibility that it performs drastically worse for those labels with less training data.

However, there exists varying discrepancies between the different ethnicities in terms of similarity in culture, which inherently affects the similarity in ingredients utilized. Thus, rather than evaluating correct predictions with a score of 1 and misclassifications with a score of 0, we can explore evaluation metrics that allow consideration of those discrepancies by implementing a range in terms of accuracy for each output. For example, for a recipe that has the correct label of '*Chinese*', a mislabel of '*Korean*' can potentially be penalized less than a mislabel of, say, '*American*', which affects the evaluation results. One way to achieve this is by constructing similarity distributions across the ethnicity labels and assigning a "score" to the prediction accordingly, which allows potential misclassifications to be penalized differently with consideration of similarity/dissimilarity between certain cuisines. An additional way to implement this is by classifying with the inclusion of confidence intervals for all classes.

As for evaluation of success beyond outperforming the baseline model, we will initially set a certain classification accuracy as standard for optimality (ideally close to 100%), which will be determined/gauged after a number of tests of our model. A high classification accuracy denotes a "successful" model. Then, we will further the evaluation by implementing precision and recall measures. With those, we can plot and analyze the F-Measure, otherwise known as the F1-Score, which by definition is 2*(precision*recall)/(precision + recall). This conveys a balance between precision and recall and allows us to evaluate the robustness of our model, which also translates to its "success".

Timeline

Key dates:

- Status report due: 3/29
- Final project due: 5/14

Projected Timeline:

- David, Alvin, Brian
    - 3/01 - 3/14 (2 weeks): Source, aggregate, and pre-process data
    - 3/15 - 4/16 (3 weeks): Create the baseline and our classification model

    - 4/17 - 4/30 (2 weeks): Evaluation & fine-tune parameters to optimize performance
    - 5/01 - 5/07 (1 week): Add extra features (recipe suggestion)
- David & Brian:
    - 5/08 - 5/14 (1 week): Write final report
- Alvin:
    - 5/08 - 5/14 (1 week): Create demo UI

Existing Data & Resources

Most of the data we use will be from datasets found online from Kaggle or other sources. We are looking for datasets with a few properties. The datasets should, at a minimum, have a section for ingredients and labels for types of cuisines. Here is a sample of a few of the datasets we are considering. The model will be implemented and evaluated using existing Python machine learning resources and libraries (scikit-learn, tensorflow, keras…).

Datasets (w/ links)

https://www.kaggle.com/hugodarwood/epirecipes/kernels

- 20,000 recipes
    - Recipe rating
    - Nutritional information (ingredients)
    - Assigned category (type of food, occasion for food, etc.)

https://www.kaggle.com/kaggle/recipe-ingredients-dataset

- 40,000 Recipes
- Data: Array of Ingredients
- Labels: Cuisine Type (Text)

https://github.com/NYTimes/ingredient-phrase-tagger

- This repo contains scripts to extract the Quantity, Unit, Name, and Comments from unstructured ingredient phrases. We use it on Cooking to format incoming recipes.

https://data.world/datafiniti/food-ingredient-lists

- This is a sample data set of ingredient lists pulled from Datafiniti's Product Database. The data set covers 10,000 different food listings and includes the ingredient list for each one.

https://eightportions.com/datasets/Recipes/#fn:1

- 250,000 recipes
- Each data point pretty much only has ingredients and a paragraph of the actual instructions, nothing else.