# Final Project Report - Recipe Generator

Jingxian Bao, Yue Wu, Weichen Zhang

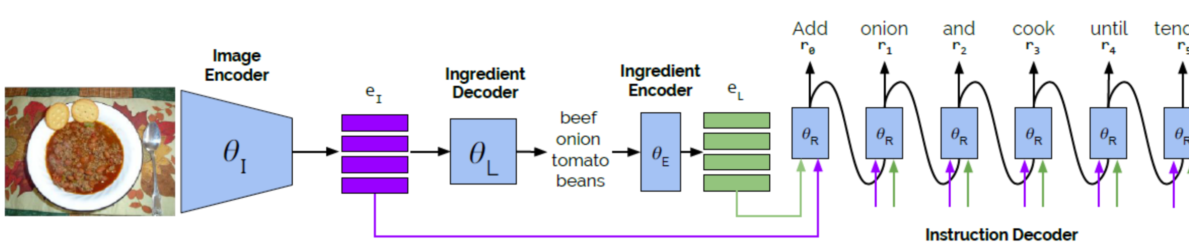
## 1. Introduction

Food is an essential part of human life. However, as the popularity of the online food delivery service continues to grow, Americans, especially the younger generations, have been cooking less. A recent research report by UBS estimated that online food delivery could dominate 10% of the food services industry by 2030. In the meantime, a research by Porch showcased that Millennials are not only cooking fewer times at home per week than Boomers and Gen X, but also consuming more frozen or prepackaged food. As cooking enthusiasts, our team decides to explore how to leverage artificial intelligence to improve the recipe generation process, in the hopes that our work may help make home cooking experience more pleasant and fun, and less daunting.

Traditionally, the recipe recommendation problem is formulated as an information search task, where a recipe is retrieved based on the ingredient similarity level from a dataset. In this project, we would like to make an AI recipe generator which learns compatibility of different ingredients from current recipes and generate its own recipe title and cooking instructions from the current ingredients users have in the fridge. Moreover, the AI recipe generator could serve as a creative aid for chefs who are seeking inspiration and innovation in their recipe creation process.

## 2. Related Works

*Inverse Cooking: Recipe Generation from Food Images*
Researchers from the Polytechnic University of Catalonia in Spain and Facebook AI Group released the inverse cooking recipe generation model on github. The goal of the model is to take an image of a cooked dish and generate the recipe of that dish. The model consists of two transformers. The first transformer takes a food image as the input and extract image features as image embeddings through the image encoder. Then the ingredient decoder will predict the list of ingredients used to cook the dish. The second transformer takes the predicted ingredient list as input and use ingredient encoder to encode the ingredient list to ingredient embeddings. Then the instruction decoder will generate the recipe title and instruction steps word-by-word using both the image and ingredient embeddings. A visualization of the model process is included below.




*Source: Amaia Salvador, Michal Drozdzal, Xavier Giro-i-Nieto, Adriana Romero. Inverse Cooking: Recipe Generation from Food Images. CVPR 2019*

*IBM Watson Recipe Generation Project*

Researchers at IBM's Thomas J. Watson Research Center are working with chefs at the Institute of Culinary Arts to build a cognitive computing system to create innovative recipes with interesting flavor combinations. The researchers set up an inspiration database with recipes from cookbooks, online recipe datasets and even the US Navy. With this large scale dataset, the researchers experimented with various transformations on the current recipes, such as ingredients switching or recipes mixing, to generate new recipes. What's even more exciting is that they are researching methods to evaluate whether the recipe would taste good by evaluating the taste-imparting molecules in different combinations of ingredients. For example, if people like a specific ingredient pairing, perhaps other pairings with similar chemical compounds would be liked too. The lead researcher Lav Varshney is excited about the potential use cases to improve school lunch menu or diet for diabetic patients. However, he does realize that people's memories and feelings around food are a crucial part of the dining experience, which are not captured within the scope of the research.

## 3. External Data Source - Recipe1M

Recipe1M is one of the largest publicly available collection of recipe data. It's a large scale dataset which contains over 1 million structured recipes and their images. Since the scope of our project does not include image related tasks, only layer1.json, which contains the structured recipe text, is used.

The recipes are stored in json format with the following key data fields:

- ID: A unique identification number for each recipe.
- Ingredients: A list that contains all the ingredients used in a recipe. The amount of ingredients needed as well as units of measurements are also included.
- Instructions: A list that contains all instruction steps to be performed to cook the dish. Each step is an element in the list. The steps are listed in sequential order.
- Partition: The recipes are partitioned in train, test and validation datasets by default.
- Title: The title of recipe.
- URL: Link to the original source of recipe.

Here's an example of sample recipe in json format:

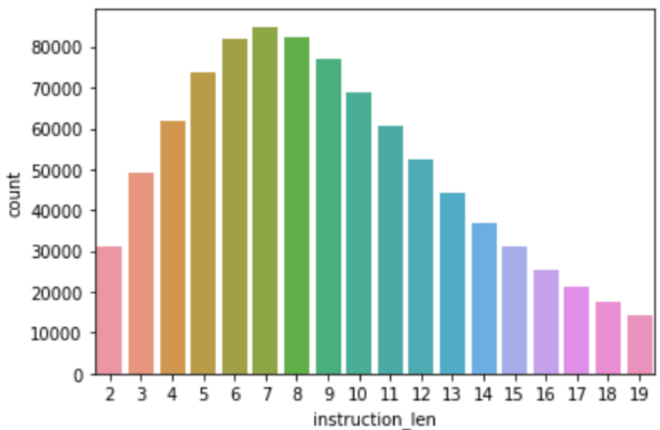

```
{'id': '000033e39b',
 'ingredients': [{'text': '1 c. elbow macaroni'},
  {'text': '1 c. cubed American cheese (4 ounce.)'},
  {'text': '1/2 c. sliced celery'},
  {'text': '1/2 c. minced green pepper'},
  {'text': '3 tbsp. minced pimento'},
  {'text': '1/2 c. mayonnaise or possibly salad dressing'},
  {'text': '1 tbsp. vinegar'},
  {'text': '3/4 teaspoon salt'},
  {'text': '1/2 teaspoon dry dill weed'}],
 'instructions': [{'text': 'Cook macaroni according to package directions; drain well.'},
  {'text': 'Cold.'},
  {'text': 'Combine macaroni, cheese cubes, celery, green pepper and pimento.'},
  {'text': 'Blend together mayonnaise or possibly salad dressing, vinegar, salt and dill weed; add in to macaroni mix.'},
  {'text': 'Toss lightly.'},
  {'text': 'Cover and refrigeratewell.'},
  {'text': 'Serve salad in lettuce lined bowl if you like.'},
  {'text': 'Makes 6 servings.'}],
 'partition': 'train',
 'title': 'Dilly Macaroni Salad Recipe',
 'url': 'http://cookeatshare.com/recipes/dilly-macaroni-salad-49166'}
```

**3.1 Exploratory Data Analytics (EDA)**

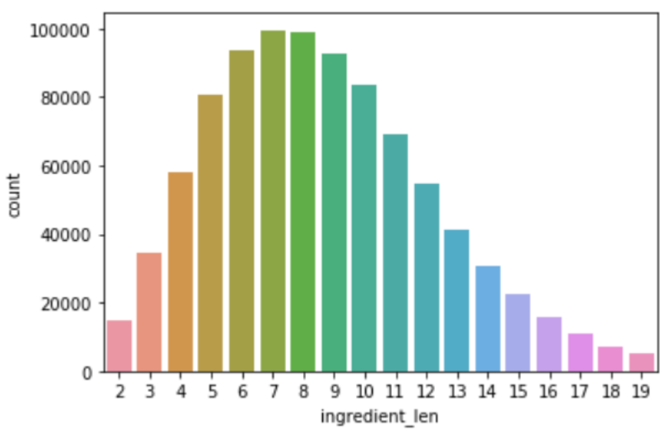
There are 1,029,720 recipes in the Recipe1M dataset. After removing recipes with less than 2 or over 20 ingredients and recipes with less than 2 or over 20 instruction steps, there are 914,208 recipes left. The dataset is separated into 639,740 recipes in the train dataset, 136,962 recipes in the test dataset, and 137,506 recipes in the validation dataset.

***Basic Data Visualization***

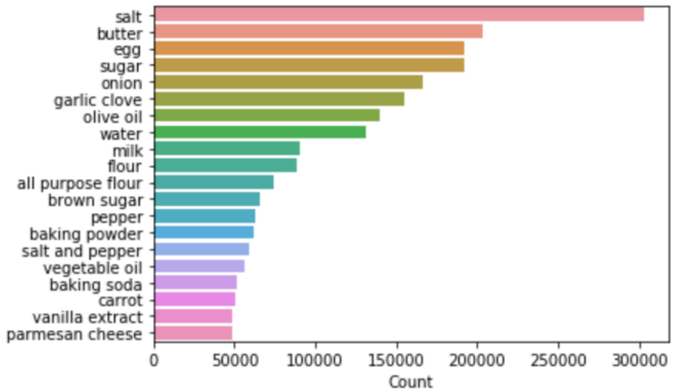
The average length of instruction is 8.914 steps per recipe. The distribution of the steps of instruction are visualized in the histogram below:




The average number of ingredients is 8.616 ingredients per recipe. The distribution of the number of ingredients per recipe are visualized in the histogram below:




The top 20 most frequently used ingredients are listed below. Salt is the most widely used ingredient that has appeared in over 300,000 recipes. It's also interesting to observe many baking ingredients, such as flour, baking powder, baking soda and vanilla extract, to be among the top 20 most frequently used ingredient list. We suspect that recipes for baked goods have strong presence within the Recipe1M dataset.

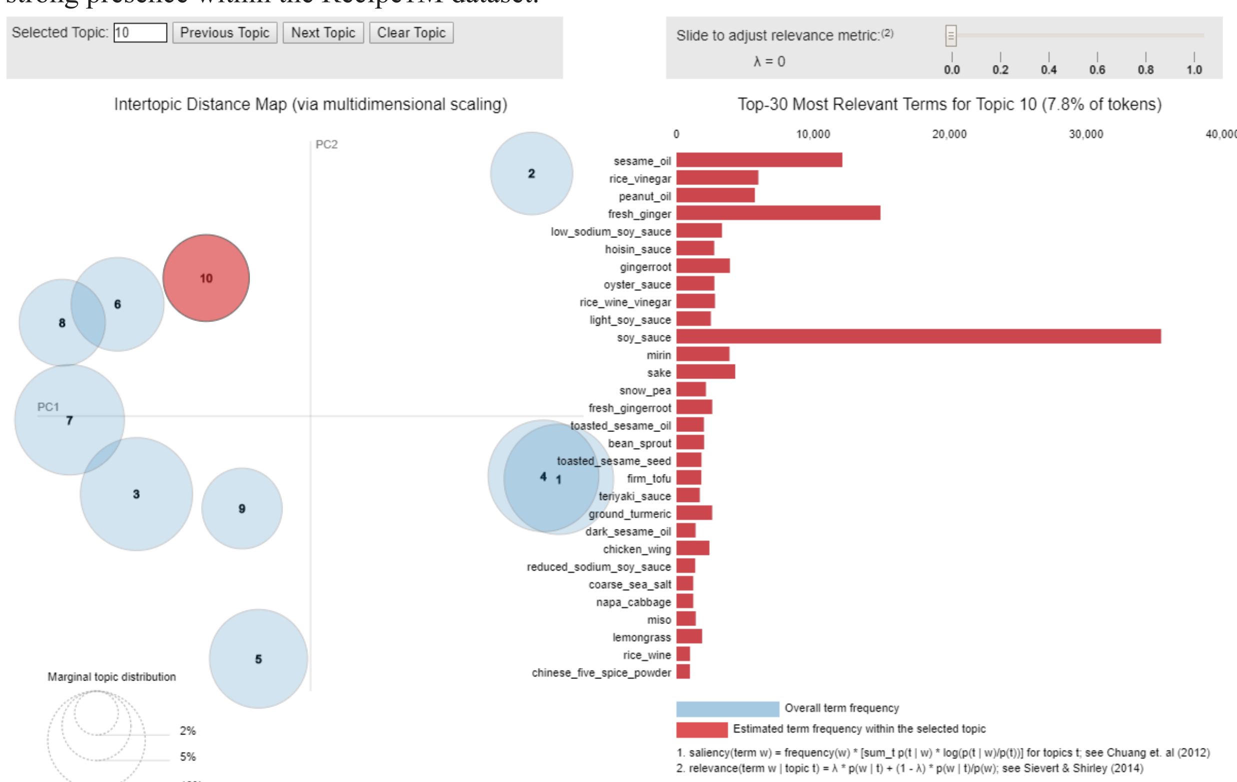



## ***LDA Topic Modeling***

Since the Recipe 1M dataset provide label to distinguish different types of cuisines, we performed topic modeling using Latent Dirichlet Allocation (LDA) over the ingredient lists to discover if certain cuisines or cooking methods are present in the dataset. Some of the most obvious discoveries from our LDA model are listed below.

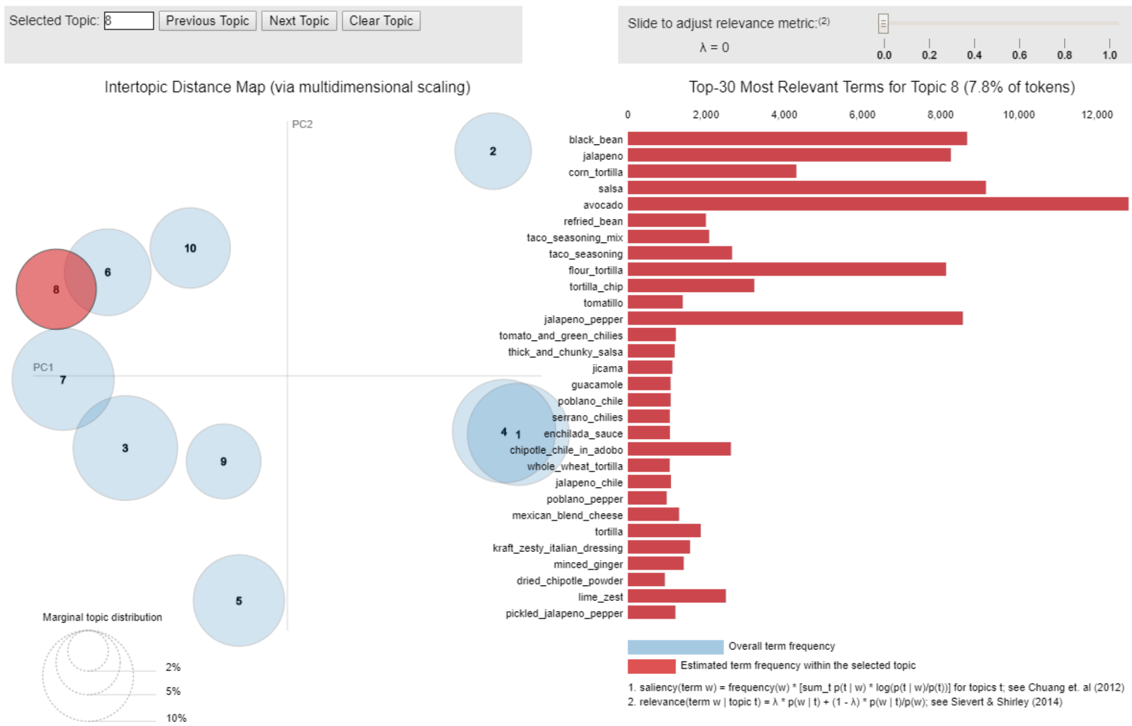
Topic 10: Asian cuisine

As the relevant metric is set to 0, the most relevant terms for topic 10 are ranked by their exclusivity in topic 10. Many signature asian ingredients, such as sesame oil, rice vinegar, hoisin sauce and soy sauce, are among the most relevant terms of topic 10. Therefore, it can be concluded that Asian recipes have strong presence within the Recipe1M dataset.



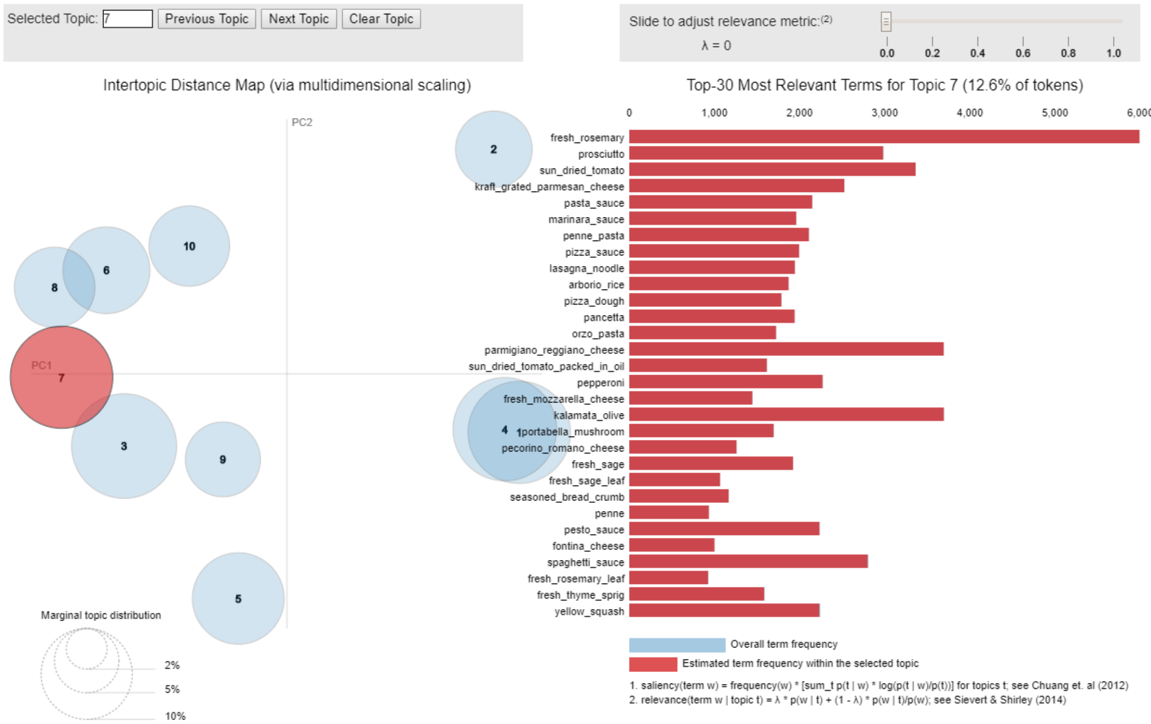
Topic 9: Mexican Cuisine

As the relevant metric is set to 0, the most relevant terms for topic 9 are ranked by their exclusivity in topic 9. Many signature mexican ingredients, such as jalapeno, corn tortillas, salsa and taco seasoning mix, are among the most relevant terms of topic 9. Therefore, it can be concluded that Mexican recipes have strong presence within the Recipe1M dataset.



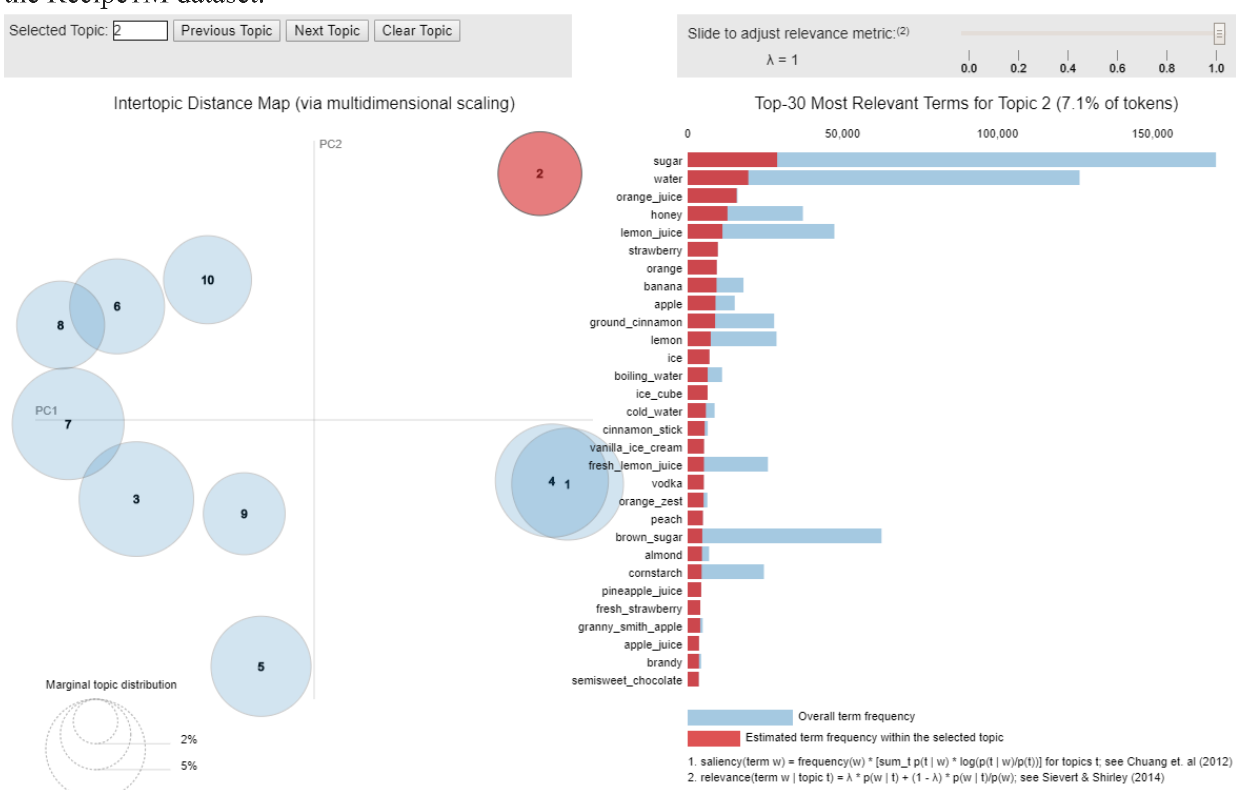

Topic 7: Italian Cuisine:

As the relevant metric is set to 0, the most relevant terms for topic 7 are ranked by their exclusivity in topic 7. Many signature italian ingredients, such as rosemary, pasta sauce, pizza dough and lasagna noodles, are among the most relevant terms of topic 7. Therefore, it can be concluded that Italian recipes have strong presence within the Recipe1M dataset.




Topic 2: Drinks, Juice, and Smoothie

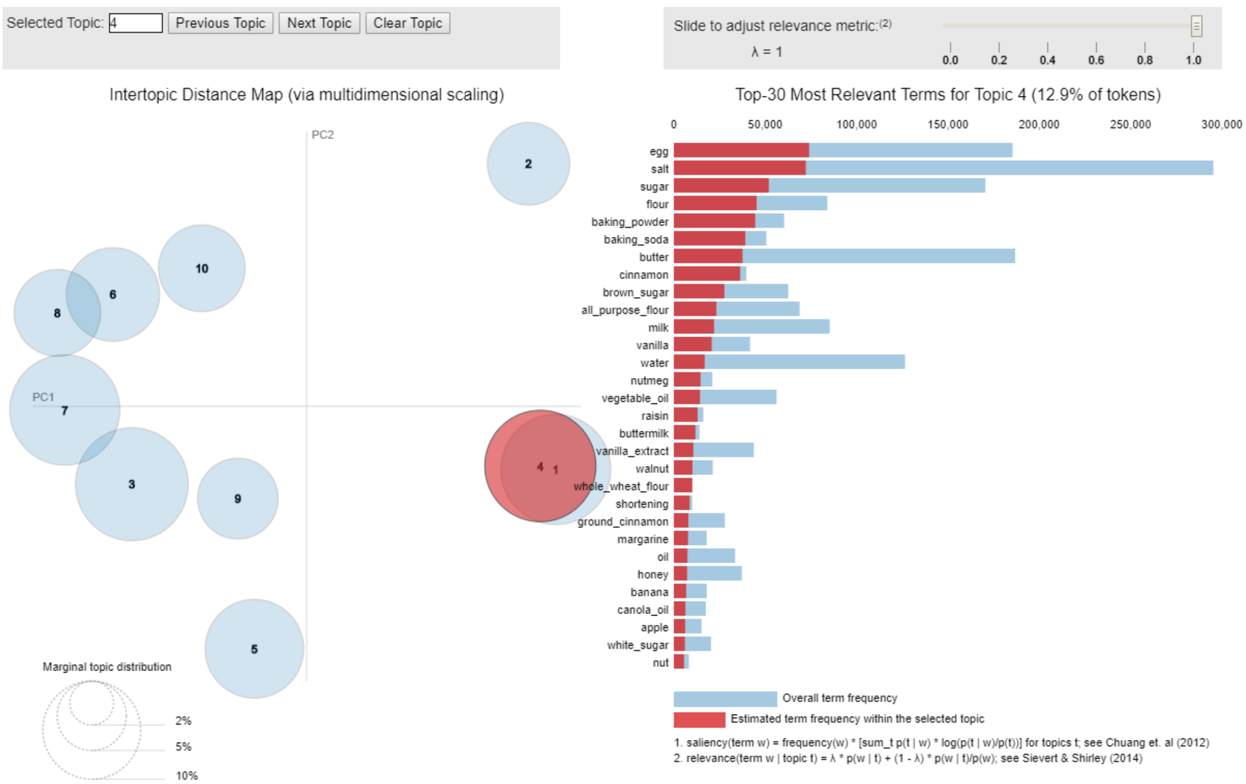
As the relevant metric is set to 1, the most relevant terms for topic 2 are ranked by their estimated term frequency within topic 2. Judging from the prevalence of fruit and liquid, such as water, vodka and lemon juice, it can be concluded that recipes for alcoholic and non-alcoholic drinks have strong presence within the Recipe1M dataset.

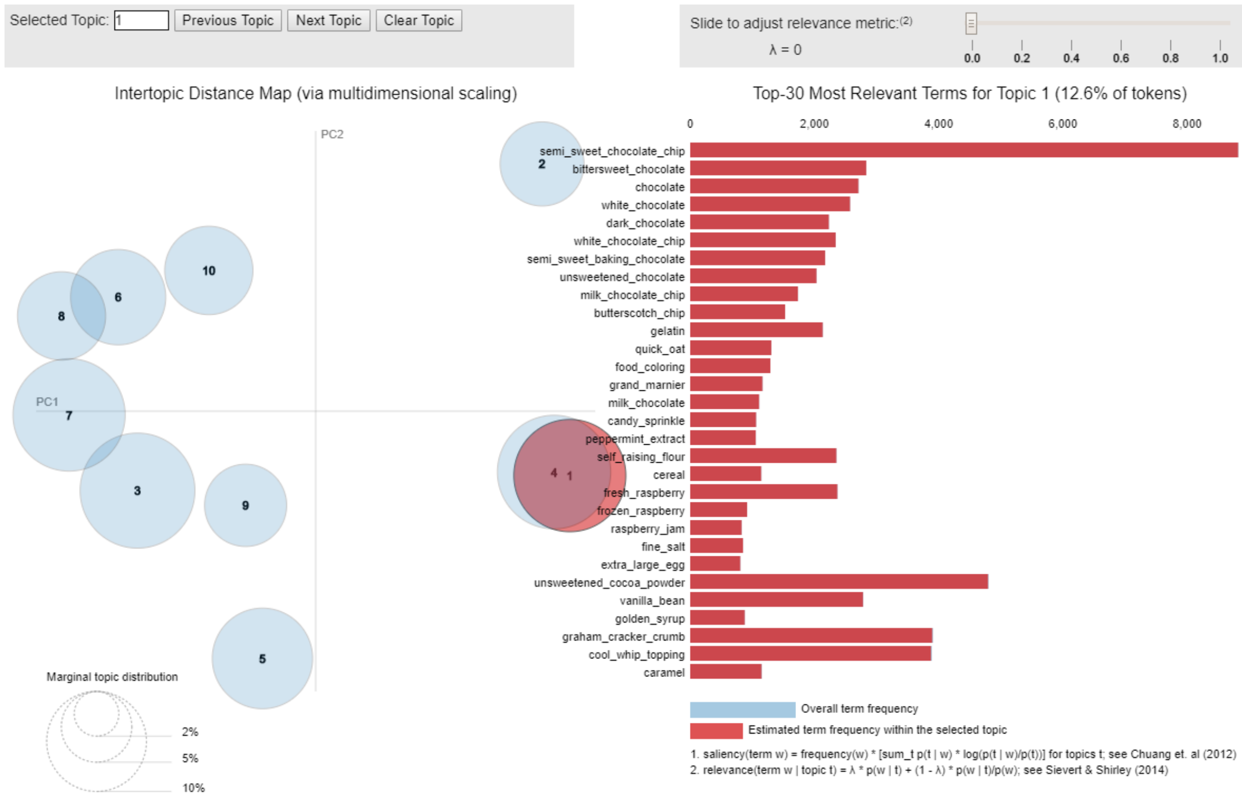


Topic 1 and 4: Baked dishes

As the relevant metric is set to 1, the most relevant terms for topic 4 are ranked by their estimated term frequency within selected topics. Judging from the prevalence of baking ingredients, such as flour, baking powder and vanilla extract, it can be concluded that recipes for baked goods have strong presence within the Recipe1M dataset.

While topic 1 shares many similarities with topic 4, we set the relevant metric to 0 to evaluate some exclusive ingredients in topic 1. The top 10 most relevant terms in topic are dominated by chocolate ingredients. Therefore, it can be concluded that chocolate flavored baked dishes have strong presence within the Recipe1M dataset.






### 3.2 Data Preprocessing

Recipe1M Dataset is obtained from website, so data is highly unstructured and the ingredients in the recipe are narrowly defined. To satisfy data consistency, we adapted following approaches before training model. Firstly, unit conversion. We try to format the same kinds of unit into a standardized form. For example, many units indicate liquid: pints, pint, pts, pt, quarts, cups, spoons. And we convert them to liter. We also format special characters, discard recipes with too few or too many ingredients to make data more structured and generalized. Moreover, for the narrowly defined ingredients, we cluster similar ingredients into the same category. Like pepper, there are 209 kinds of pepper including “red pepper”, “hot pepper”, “black pepper”. We categorize them into “pepper” when training the model. Transfer learning model could make correspondingly expansion from generalized input ingredients to specific ingredients. Besides, we add some special tokens in the recipe to separate different information such as title, ingredient, and instructions.

## 4. Our Approach

There are two major parts of our model: ingredient recommender and recipe generator. Users input several ingredients they like or in the fridge, and recommender model would select a frequently used composition from them, then delivers selected ingredients to pre-trained transfer learning model to generate recipe with instructions.

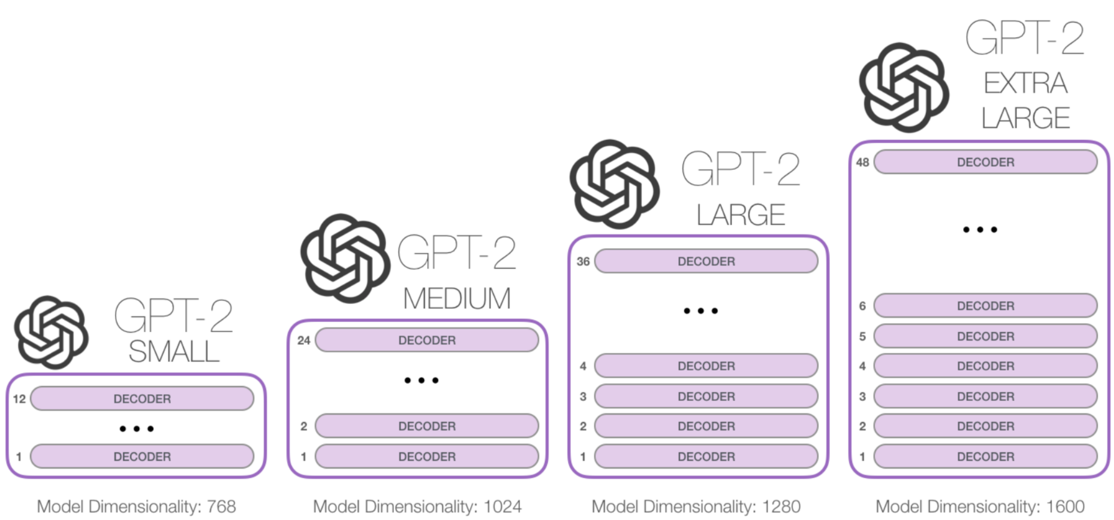
### 4.1 Ingredient Recommender

Input: ingredients from user side.
Output: a new set of ingredients for recipe generation.
After pre-processing, there are 1908 valid ingredients from training dataset. EDA shows most existing recipes have seven or eight ingredients. To generate reasonable ingredients, we select a composition of three possible ingredients that occur most frequently. Then find another three ingredients or flavours that most frequently appear with each of them according to co-occurrence matrix that derives from existing recipes. The approach allows us to generate creative dishes within a reasonable scope.

### 4.2 Recipe Generator

Transfer learning on pre-trained models have demonstrated impressive efficacy on various NLP tasks. GPT (Generative Pre-Training) and BERT (Bidirectional Encoder Representations from Transformers) are among the top pre-trained language models, yet GPT-2, the successor to GPT-2, is found to perform better in coherent text generation tasks (Wang & Cho, 2019). In light of previous research and similar projects done by other scholars, we focus on fine-tuning an OpenAI GPT-2 model to produce recipes based on the list of ingredients generated from the ingredient recommender mentioned in section 4.1.

GPT-2 is a transformer-based language model which is trained on about 8 million Reddit webpages (Radford, 2019). It is a powerful next word prediction model based on previous words in the text, consisting of a stack of decoder blocks. In fact, GPT-2 is claimed to be too powerful so that OpenAI has released its largest version (1.5B parameters) only recently on 5 Nov 2019 to control the risk of malicious use. In our work, we use the GPT-2 medium model, which consists of 24 decoder layers with a dimensionality of 1024 due to the limited computing resources of Colab.




Source: [http://jalammar.github.io/illustrated-gpt2/](http://jalammar.github.io/illustrated-gpt2/)

#### 4.2.1 Speed of Adaptation

Since GPT-2 is not pre-trained on any recipe-like content, we are interested to understand how fast the model could adapt to recipes. We measured the speed of adaptation by observing the occurrence of our special tokens (“<|startoftext|>, “<sot>”,”<eot>”,etc) in the generated text. In the first 100 steps of fine-tuning, we sample the model outputs at each step and some of the results are listed as follows.

Run 1:

```
We're not going to make that kind of trade to get me, and it's not an area where I'm very
good at. I'm a very good offensive player. I've said that all year. If we're not in the
offensive mix, then this is where I'm going to get the money.

I'm not a great blocker. My hands aren't as flexible as some guys. I mean, I'm not one of
those types who goes and pulls down, or something like that. I think for this team, I can
create some separation. I'm not a perfect blocker.
```

Run 27:

```
He scooped out the sugar while he was still stirring and used that to smooth the sugar in
and add to the creamed cream.

"Now mix with the vanilla.

He added the lemon juice, then the lemon juice to the ice cream mixture and continue
shaking and stir mixing every couple seconds after adding the cream.

"This is because I use a stand mixer.
```

Run 42:

```
<|startoftext|><soi>garlic,leek,dried thyme,garlic croutons,fresh parsley,kosher salt,fresh
ground black pepper,dissolved in 1/2 t. egg wash<eoi>

<sot>green cabbage salad<eot>

<soc>-cook kale and brussels sprouts, using 1 cooking sprayor even cooking spray for 1 few
minutes in 350f. A

...

-enjoy.<eoc><|endoftext|>
```

It is to our surprise that as early as at the 42th step, the model is able to generate the first recipe-like text. Although wordings of the generated text are still problematic, our predefined recipe structures are correctly identified with the appropriate positions of special tokens. It is also interesting to note that, on run 27, the generator is creating some narratives related to food. Since deep learning models often perform like black boxes, the sample outputs could help us interpret their learning mechanism.

#### 4.2.2 Training Approaches

Due to the limited resources we have on Colab, the session will clash easily if we try to utilize the full dataset. Therefore, two different approaches are used for model training based on smaller dataset.

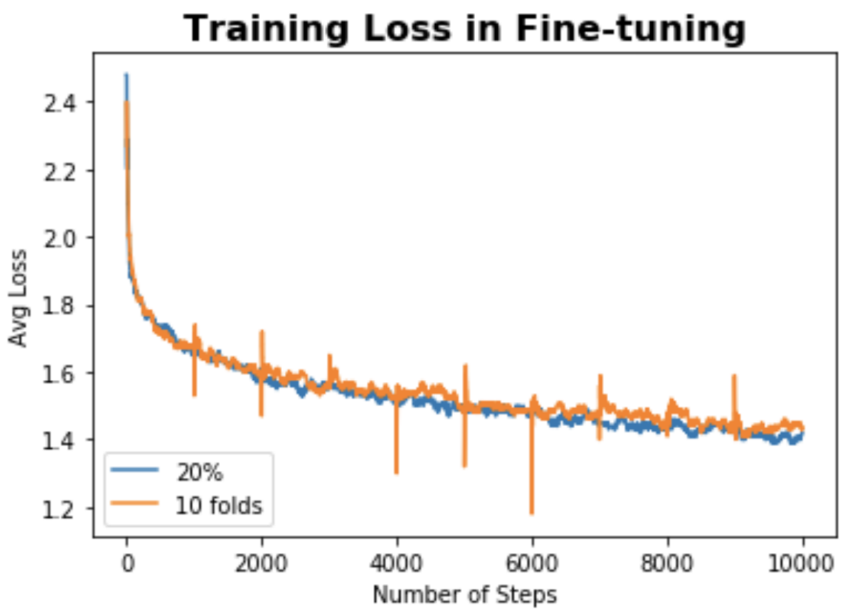
(1) 20% sampling

20% of the training data are randomly selected for a fair representation of the entire dataset we have. The model is trained on 127,948 recipes with 10,000 runs in total.

(2) 10 folds

The training data is split into 10 folds and each fold is consist of 63,974 recipes. The GPT-2 model is fine-tuned on these 10 sets of data in sequence. For example, model fine-tuned on fold-0 will be passed as the base model for fine-tuning on fold-1. In this way, the entire training set could be utilized in the training process. 1000 runs were performed for each round of training and 10,000 runs in total.

Training loss is used to compare the model performance. In GPT-2, training loss is the log error in next-word prediction. As we can observe in the diagram below, the convergence rates and ending losses are quite similar for these two approaches. However, the "10 folds" training approach is more unstable, as significant variation in error could be noted.




Moreover, recipes in the validation set are extracted as the baselines for the generated recipes. Same ingredient lists are passed into these two models and their outputs are compared with the original recipes. Here is an example below.

(1) 20% sampling

**Ingredients: all purpose flour,whole wheat flour,baking soda,salt,quick cooking oat,all bran cereal,raisin bran cereal,grape nut cereal,raisin,unsalted butter,buttermilk,granulated sugar,brown sugar,egg**

Title: bran raisin muffins

- preheat oven to 350f.
- generously butter muffin cups
- whisk together flours, baking soda, and salt in 1 large bowl, then stir in oats, cereals, and raisins
- whisk together butter, buttermilk, sugars, and eggs in another bowl, then add to cereal mixture and stir until just combined
- batter will be very thick
- spoon 1.41 l batter into each muffin and bake in middle of oven until golden and 1 tester comes out clean, 17 to 20 minutes
- remove muffins from pan and cool on 1 rack

Title: oatmeal bran muffins

- in 1 large mixing bowl combine all ingredients except raisins and stir to combine.
- -fill pans with muffin paper cups.
- -preheat 350 degrees f. in 1 large bowl, combine dry ingredients and stir to combine.
- -set aside.
- -in another large mixing bowl, scald butter with buttermilk.
- -add butter, sugar and eggs and stir just until moist crumbs form.
- -spoon mixture onto muffin paper cups, making 15 muffins.
- -bake at 350 for 18 minutes

As highlighted in the text, the model is able to tell that bran muffins could be made using the given ingredients. Moreover, it is able to provide key steps, such as to preheat to 350 degrees, mix ingredients, spoon onto paper cups, and bake for 17-20 minutes, to prepare the muffins.

(2) 10 folds

Title: bran raisin muffins

- preheat oven to 350f.
- generously butter muffin cups
- whisk together flours, baking soda, and salt in 1 large bowl, then stir in oats, cereals, and raisins
- whisk together butter, buttermilk, sugars, and eggs in another bowl, then add to cereal mixture and stir until just combined
- batter will be very thick
- spoon 1.41 l batter into each muffin and bake in middle of oven until golden and 1 tester comes out clean, 17 to 20 minutes
- remove muffins from pan and cool on 1 rack

Title: bran bran bran muffins

- warm oven to 220 degrees.
- set oven rack to middle position and position 1 rack in centre of oven.
- in 1 large bowl, add flour, wheat flour, baking soda, salt and oats.
- in 1 food processor, pulse oats, bran, raisins, oats rye dry cereal, grapes, raisins, grapes, raisins, raisins, raisins, grapes, raisins, grapes, raisins, grapes, grapes, raisins, dough will be crumbly, and will break up .
- ...

However, in the text generated by the "10 folds" model, some repeated phrasings are observed. Since only 10% of the corpus is used in the last round of fine-tuning, the model may have less tokens to generate the text. This smaller vocabulary might also contribute to the large variations in the training loss. Therefore, based on the graph of train loss and human judgement, 20% sampling is a better approach to fine-tune the GPT-2 model on recipes.

**Future Improvements**

First of all, only 20% of the data and the medium GPT-2 model is used in our work for training. If time and computing power allows, more coherent recipes are highly expected by training on the larger GPT-2 model.

Moreover, we could leverage on the cuisine classification to build more specific models on each type of cuisine. For example, we could extract all pizza recipes and fine-tune a pizza-expert AI chef. Since the recipes for a specific type of cuisine are more standard, we expect the GPT-2 to learn the recipe patterns faster and generate text with higher quality.

For recipe evaluation, we could use crowdsourcing or focus group to get user feedback on generated recipes. The users will be presented with both the real and AI generated recipes and they will rate both recipes on ease to understand, feasibility and creativity. We could also post both the real and AI generated recipes on the online food and cooking social networking platforms such as Allrecipes for a period of time and compare user ratings and feedback.

**Lessons Learnt**

***Did anything surprise you?***

Since GPT-2 was not pre-trained on recipe-specific text, we were initially concerned that the generated recipes would bear more similarities to the GPT-2 pre-trained text. We are surprised about how few steps are required for the generator to learn the recipe structure.

***Was anything especially hard or easy?***

Even with free TPU computing resources provided by Google Colab, running a pre-train model on over just a portion of the 1 million recipe text took us 3 hours on average, not mentioning training text generation model from scratch.

**Distribution of Labor**

Jingxian Bao: Data preprocessing, GPT-2 and Evaluation
Yue Wu: Recommender
Weichen Zhang: Exploratory Data Analysis, LDA topic modeling, experimentation with inverse cooking model and BERT

**Appendix**

Code Organization

Code folder: includes all codes for recipe generator project

- 1.build_vocab.ipynb: Data preprocessing
- 2.EDA.ipynb: EDA and LDA topic modeling
- 3.Recommender_new.ipynb: Ingredient recommender
- 4.gpt2-355M_adaptation_test.ipynb: GPT-2 model for adaptation test
- 5.gpt2-sample5-355M.ipynb: GPT-2 model with 20% sample
- 6.gpt2-slide10-355M.ipynb: GPT-2 model with 10-fold

Data folder
- Recipe1m_train.pkl: processed training data
- Recipe1m_test.pkl: processed testing data
- Recipe1m_val.pkl: processed validation data
- Recipe1m_vocab_ingrs.pkl: ingredient vocabulary with clustering

Model folder: includes pre-trained LDA models and co-occurrence matrix used for recommender. Fine-tuned GPT-2 models are too large to be included in the submission, so please approach any of us if GPT-2 models are required for the assessment.

**Reference**

Amaia Salvador, Michal Drozdzal, Xavier Giro-i-Nieto, Adriana Romero. Inverse Cooking: Recipe Generation from Food Images. CVPR 2019

A.Wang, K.Cho, BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model, (2019). http://arxiv.org/abs/1902.04094 (accessed March1, 2019).

Cooking Nightmares. (n.d.). Retrieved from https://porch.com/resource/cooking-nightmares.

Hanbury, M. (2018, June 25). Millennials are cooking less and less, and it could cause a crisis for America's biggest food companies. Retrieved from https://www.businessinsider.com/millennial-cooking-habits-threaten-general-mills-kraft-heinz-2018-6.

Radford, A. (2019, November 6). Better Language Models and Their Implications. Retrieved from https://openai.com/blog/better-language-models/.

Ross, V. (2013, May 31). Creating Recipes with Artificial Intelligence. Retrieved from https://spectrum.ieee.org/computing/software/creating-recipes-with-artificial-intelligence.