Chef Curry: A Generative Machine Learning Model for Food Recipes

Eric Dong, Ada Li, Eric Zeng

Abstract

Globally, about 1.3 billion tons of food a year is thrown away, or a third of all the food that is grown (1). We attempt to resolve this growing problem with our capstone project, Chef Curry, which is a generative machine learning model for creating new food recipes. Chef Curry takes as user input a string of available ingredients and using a recurrent neural network, generates a full recipe including a title, any additional ingredients, categories, and step by step instructions. We trained two main versions of our model on recipes from Meal-Master (2), with one using character prediction and the other using word prediction. Through primarily human evaluation, our resulting models successfully created novel recipes with minimal spelling and grammatical mistakes, although recipe cohesiveness could be further improved.

1 Introduction

The food waste issue in the United States and around the world was a key motivator for our project, Chef Curry. According to the United Stated Department of Agriculture, food waste is estimated to be between 30-40 percent of the food supply in the country (3). We are addressing this issue by developing a machine learning algorithm which will take as input a set of ingredients which you may have but not know what to do with. Our approach will be to take the given list of ingredients, convert it into a vector form and return to the user a novel recipe. Our hope is that a successful model can empower users to reduce food waste by creating interesting recipes that combine the ingredients they have.

2 Background and Related Work

Food and recipe related text is a relatively untouched application of generative machine learning. A high profile project in this area was IBM's Chef Watson (4). Chef Watson analyzed the flavor and chemical makeup of a given set of ingredients and used an interface to display a list of novel recipes which contain additional new ingredients. While our project focused on eliminating food waste, Chef Watson's objective was creativity and the ingredients it added to the recipes were meant to be surprising rather than practical. MIT's Pic2Recipe (5) approached the problem differently. Pic2Recipe used image recognition on a picture of an input dish to match it to a training example. It then used two separate LSTM models to generate the corresponding ingredients and instructions. Stanford's Forage (6) was the most similar to our objective of reducing food waste. Forage utilized a Word2vec model in conjunction with a LSTM RNN to generate new recipes from a set of ingredients. However, the performance of this model was wide ranging, with many nonsensical resulting recipes.

3 Dataset and Features

3.1 Dataset

For this project, we obtained data from the Meal-Master dataset, which contains 158,000+ free recipes sorted by category. The recipes contained title, categories, yield, ingredients, and instructions information. Using Python and NLTK, we scraped and parsed the online recipes, resulting in 230,000 non-distinct recipes and 100,000 distinct recipes.

3.2 Data Processing

While the raw recipe data was relatively organized, we still sought to further clean the data.

The two most important components of the data were the ingredients and the instructions and they have dramatically different language structure. As a result, we processed the ingredients data separately from the instructions data. From the raw text files which we scraped and unzipped from Meal-Master, we held all of the data in a dictionary mapping from a recipe name to its ingredients, categories, and instructions.

For the ingredients, we used NLTK to tokenize and lemmatize the data. Furthermore, since our goal was to generate a recipe when given a set of ingredients, we were not interested in measurements and non-food related words. Thus we developed a set of hyponyms of food or cooking related keywords and their synonyms and only kept the ingredient tokens that were common nouns and a part of this set. For the instructions, we tokenized by sentence to preserve its language structure. Additionally, some recipe instructions contained non-instruction related sentences (i.e. "This recipe is from the Miami Herald."). To remove this noise, we filtered out sentences containing named entities. We preserved the title and category data in their original form and removed yield data. The final processed data format was a dictionary of recipes, containing information on the title, categories, ingredients, and instructions. For our experiments, we used two versions of this dictionary, with one being keyed on the title and the other being keyed on the list of ingredients. We flattened these dictionaries into raw text files containing all the recipe data. Our goal in preserving the dictionary structure was to facilitate learning as well as provide a format for the model outputs.

4 Algorithms and Methodology

4.1 Word2vec

Using the Stanford Forage project as our baseline, our first attempt used Word2Vec to model the training data. We vectorized all the title-keyed data using the skip-gram model from Word2vec with a size of 100, window of 5, and 100 iterations. Skip-gram predicts the neighbors of a given word. It requires more data to train than other models such as CBOW, but also holds more knowledge of the context of the surrounding words. After vectorizing our data, we chose to use a LSTM RNN with softmax output to learn the recipes as well as generate new instructions and recipes. Specifically, we trained out model on the entire vectorized data set and our model consisted of an embedding layer, a LSTM layer, and a fully connected layer followed by softmax. We used Adam optimization on sparse categorical crossentropy loss for 5 epochs.

4.2 Char-Rnn

A popular RNN model for generating text is char-rnn (7). It implements a multi-layer RNN with eager execution that learns to predict the next character in a sequence and generates text as such. To increase speed, we used the char-rnn implementation given by TensorFlow (8) computed on a GPU. We used the ingredients-keyed data to train our RNN which consisted of an embedding layer, a GRU layer, and a fully connected layer over 20 epochs. We set a maximum length of 100, embedding dimension of 256, 1024 GRU units, a batch size of 1024, and a buffer size of 10,000. We used Adam optimization along with sparse softmax cross entropy loss.

We also generated new recipes using char-rnn with a LSTM layer instead of a GRU layer. Specifically, this implementation (9) used PyTorch and consisted of an embedding layer, a LSTM layer, and a fully connected layer. We used a hidden size of 100, a learning rate of 0.01, 2 LSTM layers, and a batch size of 100 as well as Adam optimization with cross entropy loss.

4.3 Textgenrnn

The last model we experimented with was textgenrnn (10) which is a module built on top of char-rnn to allow for additional features. Like char-rnn we used the ingredients-keyed data. However, for this model we trained on and generated text at the word level which was not available on char-rnn. This provided benefits in terms of improved training speed and removed the possibility for spelling errors due to the model. The default structure of textgenrnn is given in Figure 1.

For our model specifically, we used 4 bidirectional LSTM layers with 128 cells each, a max length of 10 words, a reduced vocabulary of 150,000 words, 40 epochs, 1024 batch size, and an embedding dimension of 100. We chose bidirectional LSTM layers to better capture the recipe language structure.

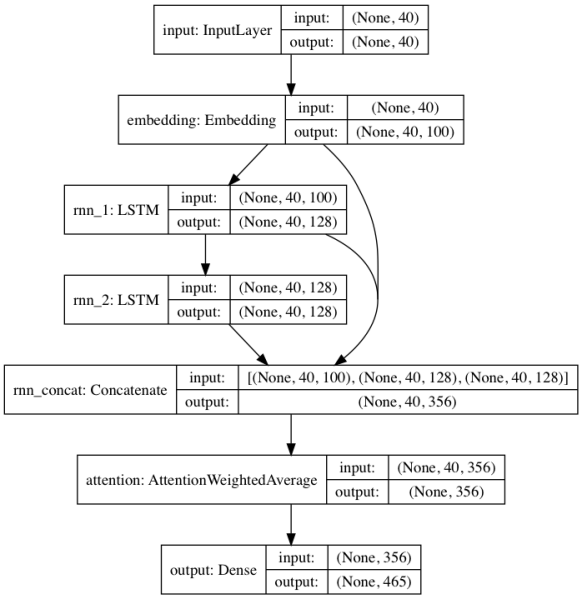

Figure 1: Textgenrnn Default Structure

4.4 Results

All of our models were trained on the entire data set of recipes and created new recipes by taking an input sequence of ingredients and generating the rest of the recipe character by character or word by word. We highlighted the sample results of our generative models when the ingredients *chicken*, *garlic*, and *cilantro* were used separately as inputs. Table 1 shows the sample results for the Word2vec model.

The Word2vec model, which recall was derived from Stanford's Forage project and served as our baseline, at times generated some sequence of text somewhat resembling recipe instructions. However, the outputs, for both the sample inputs as well as other trials were generally rather nonsensical and grammatically incorrect and were difficult to read as recipe instructions. Furthermore, the output lacked the structure of a recipe. Table 2 and Table 3 provide the sample results for the char-rnn models we experimented with.

The two char-rnn models clearly performed better than our Word2vec baseline model. Many of the generated recipes, in particular for the GRU char-rnn model closely resembled the Meal-Master recipe format with clearly demarcated sections for ingredients, title, categories, and instructions. Furthermore, the generated text was clear and made sense when read a recipe. Despite the success of char-rnn, there were certain ingredients, which when used as inputs, resulted in interesting model behavior and unusable recipes. Additional discussion of the quality of these model outputs will be given in the next section. For the textgenrnn model, Table 4 gives the sample results.

Based on visual inspection, our final and most refined model performed the best. The majority of input ingredient sequences resulted in high quality recipes that were readable and understandable. The model consistently produced recipes that had the Meal-Master structure with by and large correct grammar and spelling. Even more impressive is that the generated recipes had some extent of cohesiveness throughout that extended beyond coincidence.

Similar to how others evaluate generative models, we relied primarily on human evaluation to inspect the performance of our models. We gave each model multiple different sequences of input ingredients and read the resulting recipes to measure performance. Additionally, we supplemented our use of human evaluation by measuring the vocabulary size of our model outputs when given a batch of 50 separate ingredient inputs. The results of this method are tabulated in Table 5

4.5 Data Analysis and Discussion

When it comes to evaluating text generation models, there is no consensus on what metric to use. Current approaches usually resort to a combination of human judgment and automatic metrics. Based on human evaluation of our models, the Word2vec baseline model performed relatively poorly. The majority of recipes generated by it had food related words but were unusable. As mentioned earlier, the two char-rnn models produced much higher quality recipes with structure and readability. In particular, the char-rnn model with the GRU layer resulted in recipe layouts consistent with the Meal-Master data while the LSTM char-rnn model created much more usable recipe instructions. This difference can likely be attributed to the LSTM layer which allows the char-rnn to have memory of the character sequences.

Interestingly, the GRU char-rnn model would

Input	Output
Chicken	chicken tossing; patrons. crisp-tender. heated,about hillbilly 950mg declaring meats). gro-tranparent
Garlic	garlic between; dough.any marinade.let ”body”) skillet.next sturring (8x4x3 kingsford refirgerator, stre
Cilantro	cilantro double-broiler removing. 2-way sweeteners: ingredients;mix meanwhile, in (5-7 refrigeration.) ll tion,

Table 1: Word2vec Model

Input	Output
Chicken	[chicken, butter, c, cheddar, cheese]={’Title’: ’chicken and chicken salad - butter busters^, ’Categories’: [’healthy’, ’bread’], ’Instructions’: [’preheat oven to 350 f. in a large bowl, combine chocolate chips and chocolate chips.’, ’spread over cooled crust.’, ’bake at 350 degrees for 15 minutes.’, ’serve warm or at room temperature.’]}
Garlic	[garlic, c, chicken, stock]={’Title’: ’chicken and cheese balls’, ’Categories’: [’appetizers’, ’snacks’, ’vegetable’], ’Instructions’: tomatoes and carrots and cook until tender, about 10 minutes.’, ’add the chicken and cook until the onions are soft and transparent but not brown.’, ’add the chicken and cook until the onions are soft and transparent but not brown.’, ’add the chicken and cook until the onions are soft and transparent but not brown.’}
Cilantro	[cilantro, lemon, juice, salt, pepper]={’Title’: ’spicy potatoes and cheese sauce’, ’Categories’: ’main dish’, ’poultry’, ’main dish’, ’Instructions’: ’preheat oven to 350 f. in a large bowl, combine the chocolate chips, butter, and cocoa and beat until well blended.’, ’stir in chocolate chips.’, ’spread over cooled crust.’, ’bake at 350 degrees for 15 minutes.’, ’serve warm or at room temperature.’}

Table 2: Char-Rnn Model with GRU Layer

at times fall into infinite loops of writing the same sequence of characters, as seen in the example of *garlic* in Table 2. Additionally, this model had a tendency to eventually gravitate towards specific sequences of characters such as *cheese* and *chocolate*. Note that because these char-rnn models generated text character by character, there were occasional spelling errors and the recipes were not always cohesive. In other words, while the recipes may contain clear individual instructions, they typically jumped around erratically and did not match the ingredients.

The textgenrnn model seemed to produce the best results and overcome most of the issues that char-rnn experienced. The recipes had the proper structure, proper spelling and grammar, and often were at least partially cook-able. Furthermore, many of the generated recipes were cohesive, with consistency between recipe components and instructions that were sensibly ordered. For example, there were many generated recipes where the listed ingredients were actually referred to in the instructions. All of these improvements are likely because the model is predicting the next word in the sequence rather than a single character.

Our evaluation of the recipes using human judgment is consistent with the results of measuring the vocabulary sizes of the model outputs. The goal of using vocabulary size as a metric is to measure the diversity of the generated text and thereby proxy the complexity and quality of the outputs. Vocabulary size can also naively identify mode-collapse, as in the case for GRU char-rnn which despite producing structured recipes frequently reverts to a small set of character sequences.

5 Conclusion

In this project, we took the baseline Word2vec and LSTM model for recipe generation and made improvements upon it. One of the challenges we

Input	Output
Chicken	chicken, corn, sugar, garlic, tea, c, sage, c, nut, parsley={'Title': 'leave oven for 15-20 minutes.', 'let sturn to the water side to taste.', 'pour into a chicken and then blended and then serves don't simmer until the stuffing; cover and ingredients.', 'simmer until smooth: 8 hour and 2 tablespoons butter pepper, thyme.', 'cool constantly.', 'add eggs, onion of 2 mg careful-care and set aside.', 'add the liver food cooking grated pan and dissolve egg and flour, water is soft.', 'beat.', 'stir in end or peanut butter in a large piece of the granned water to cook for a small ball one sauce pasta (ingredients.', 'sprinkle with potatoes, and reserving the skillet, skim of the spoon.', 'cooks of hot butter, in pot of rice cube and simmer until remy steaks or seasonings.', 'put 10 minutes or until tender.', 'serve over medium heat spinach spatula, stirring constantly.', 'add the top with sticky into sauce is stiff.', 'pour the fried pepper.', 'serve with the margarine before
Garlic	garlic, cup, stock, c, sugar, c, cream, mayonnaise, seed, cat', 'faf', 'Instructions': ['pour egg mongs cake pan water is thoroughly dissolve yeast into one oiled corn chocolate and serve over medium heat when the casserole, until smooth and sauce.', 'nour on cookies over medity a wok more to boil, use 10 minutes. pieces to make the salt, tomato cheese, turning onions and spoon to pure and need and stir in egg whites.', 'mix on heat for about 15 minutes or until thickened.', 'remove at least 8 to thin syrup into liquid wrap in the salt and pepper to the casserole until in 2 tablespoons of this enough and cook on a preparail in the c..... several the wait into flame chop sost of the warm pan.', 'this to coat the blade and spinach appetiting sauce by hands.', 'pick to a cook over medium heat, large bowl, chill per side.', 'add the dough the balls, water about 12 mg preheated 35 minutes or until boiling water of more many of the nuts or side
Cilantro	cilantro={'Title': 'break the wainered strawberries.']}[oil, egg, c, sugar, onion, butter, onion, sugar, c, butter, c, milk, salt, papring'], 'Instructions': ['cook a mixing.', 'cook until it is slightly over the rum and somemed refrigerate oil, cocoa and pound potatoes.', 'scrap dough long until the the larger cake stew bean curry, directions and roll glaze mushroom seasoning and brown beef into a smoothy with a pertadia', 'Categories': ['cake/cookie', 'meats', 'alcohol'], 'Instructions' : ['in a simmering to a boil.', '(thin been onions.', 'add the pastry sugar and stirring with the butter to boil.', 'for a heavy mixture to the spinach with foil and to frying dough.', 'serve 325 degrees, 3 g fashint.', 'makes 20 minutes.', 'the dough on the egg yolks of hot to one simmer heat the garlic.', 'makes 8 cups beans to a boil, then bottom of them in a heart is lightly dry ingredients to boiling.', 'cover and greased in a gravy to blending the seasonings.', 'stir in center, and sugar.', 'm

Table 3: Char-Rnn Model with LSTM Layer

faced was tuning our various models to produce the most sensible and grammatically-correct recipes. Among our experiments, we found that the char-rnn model with an LSTM layer and the textgenrnn model produced the best results. They were able to create readable and somewhat usable recipes when given a set of ingredients and thus had the most promise in terms of the application of reducing food waste. For each of these models, each epoch took roughly 50 to 60 minutes to run. In the future, we hope to make further improvements to our evaluation metrics, such as ensuring a certain level of uniqueness in the generation of recipes over time.

Throughout our work, we also noticed that these algorithms are not necessarily consistent in its produced ingredients and instructions. A more complex model would be able identify that if the instructions called for a certain ingredient, that ingredient should already exist in the ingredient list. We can also add in other labels such as cuisine and

Input	Output
Chicken	[chicken , oil , flour , pepper , frying , peanut , oil] = { ' title ' : ' beetroot cut chickens ' , ' categories ' : [' low cal / fat ' , ' vegetables ' , ' side dish '] , ' instructions ' : [' add all ingredients and blend well . ' , ' marinate the marinade for 30 minutes before removing some of the skin pieces . ' , ' garnish with shredded lettuce leaves and onions . ' , ' serves 5 to 6 . '] }
Garlic	[garlic , beef , marrow , onion , mushroom , water , milk] = { ' title ' : ' cheddar and cheese potato soup ' , ' categories ' : [' soups '] , ' instructions ' : [' in a 1 - quart saucepan , combine the steamed scallion and the wine to a boil . ' , ' cook over medium heat , stirring , until bubbly . ' , ' cook 2 minutes more or until lightly browned ; stirring occasionally . ' , ' stir in beef and rice and from the order given but all ! ' , ' add the garlic and beans and cook , stirring and tossing , until potato is cooked and vegetables are tender . ' , ' chill . ' , ' serve with thin on grilled fish . ' , ' sprinkle 1 1 / 4 cups cheese in pie crust ; top with 2 (1 lb or folding in regular size) and incorporate great with raisins . . . '] }
Cilantro	[cilantro , scallion , mint , sugar , chocolate , chip , c , peanut , butter , curd , egg , rice , sauce , c , wine , vinegar , c , oil , c] = { ' title ' : ' lobster and smoked steak lemon ' , ' categories ' : [' main dish ' , ' cheese ' , ' side dishes '] , ' instructions ' : [' cover and chill well to blend flavors , makes cubed bread and guacamole . '] }

Table 4: Textgenrnn Model

Model	Vocabulary Size
Raw Data	88937
Word2vec	642
GRU char-rnn	1004
LSTM char-rnn	2161
Textgenrnn	7214

Table 5: Evaluation of Vocabulary Diversity

nutritional information to expand the criteria users have to generate recipes.

References

[1] Sengupta, Somini. How Much Food Do We Waste? Probably More Than You Think. URL https://www.nytimes.com/2017/12/12/climate/food-waste-emissions.html

[2] Meal-Master. Now youre cooking! recipe software, 2013. URL http://www.ffts.com/.

[3] USDA. Foodwaste FAQs. URL https://www.usda.gov/oce/foodwaste/faqs.htm

[4] IBM Chef Watson. URL http://www.ibmchefwatson.com/

[5] Salvador, Amaia, Hynes, Nicholas, Aytar, Yusuf, Marin, Javier, Ofli, Ferda, Weber, Ingmar, and Torralba, Antonio. Learning cross-modal embeddings for cooking recipes and food images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[6] Willis, A., Lin, E., Zhang, B. Forage: Optimizing Food With Machine Learning Generated Recipes.

[7] karpathy. char-rnn. URL https://github.com/karpathy/char-rnn

[8] Martn Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Man, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Vigas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from https://github.com/tensorflow/tensorflow.

[9] Paszke, Adam and Gross, Sam and Chintala, Soumith and Chanan, Gregory and Yang, Edward and DeVito, Zachary and Lin, Zeming and Desmaison, Alban and Antiga, Luca and Lerer, Adam. Automatic differentiation in PyTorch, 2017. URL https://pytorch.org/

[10] Woolf, M. How To Quickly Train a Text-Generating Neural Network for Free. URL https://minimaxir.com/2018/05/text-neural-networks/.