NLPChef - An NLP model for cooking recipe creation

Tim Holthuijsen, Sergio Kazatzidis, Emilia Chammas

1. Abstract

This paper first explores the use of the GTP-2 model in cooking recipe creation. Then, and mainly, it presents a custom model for recipe creation based on input ingredients. The GTP-2 model, based on auto-regression, is explained, in order to elaborate on the need for a more customizable model. To train this custom model, a dataset of scraped recipes from three food websites is used. The model is generated based on Sampling and Top-K sampling with transformers. The GTP-2 model does not produce very coherent recipes, although the topic of cooking is clear. The custom model produces more coherent sentences, although the method is sometimes unusual. This can be improved by filtering the dataset more carefully. Overall, the custom model performs better on cooking tasks than the GTP-2 model.

2. Introduction

In many cultures, food is not only a means of survival, but an integral part of social life. Whether at family dinners or in new trend-restaurants with friends, the demand for new, innovative recipes constantly grows. However, this is a time-consuming task, based on trial and error. To make this process more efficient, we propose to apply Natural Language Processing (NLP) to the creation of recipes. NLP is a Machine Learning technique used to understand and analyze human language. As language is constantly used in all aspects of life, the applications of NLP are close to limitless. Common uses include hate-speech detection, spam filters and translation. These can be expanded to areas seemingly disconnected from technology. One of those is creating recipes. In the following, OpenAI's transformer-based GPT-2 model will be applied to create and evaluate an assortment of recipes generated by an Artificial Intelligence. Furthermore, an own model was developed, taking ingredients as input, and outputting a possible recipe with those ingredients. This aims to minimize waste and enhance creativity while cooking.

3. Related Work

There have been several attempts to process recipes. Common approaches include bag of words, complex semantic representations and dependency tree data structures. Using bag of words, Ahn et al. (2011) created a network based on flavors, Jain et al. (2015) expanded this to include cultural differences in eating behaviour. Nedovic (2013) assessed recipe co-occurrences using latent Dirichlet allocation and deep belief networks.

Using semantic representations, Tasse and Smith (2008) developed a language to annotate recipes. In Mori et al.'s study (2014), graph nodes represented ingredients, tools and actions. Neither model reported significant success.

Jermsurawong and Habash (2015) suggest an ingredient- instruction dependency tree representation of recipe structure. This is called SIMMR (Simplified Ingredient Merging Map in Recipes). It focuses on the high-level flow of ingredients without individual modeling the semantics. It reached a parsing accuracy of 93.5%. Garrido-Merchán and Garrido-Merchán (2018) used Bayesian Optimization to reduce noise of datasets.

Overall, a wide range of approaches has been applied to processing cooking recipes with varying recipes. None of those take ingredients as input, creating a gap that this paper is aiming to fill.

4. Data Collection and Description

To train the custom model, a dataset by Ryan Lee in 2017, who scraped 125164 recipes from three food websites was used. Those websites include Foodnetwork.com, Epicurious.com and Allrecipes.com. The data was originally split by website, but combined in the pre-processing. The recipes contain a title, a list of ingredients/measurements, instructions, and a picture of the dish. Originally, the data was in a dict-of-list-of-dict format, which is not convenient to work with. Hence, the data was converted to a pandas dataframe. This removed 517 empty values and left us with 124647 recipes. Additionally, one out of the three datasets used contained the word "Advertisement" frequently. This was removed during the pre-processing. The data was added to the dataframe, according to the categories "title", "ingredients" and "instructions":

	Title	**Ingredients**	**Instructions**
0	Slow Cooker Chicken and Dumplings	[4 skinless, boneless chicken breast halves , ...	[Place the chicken, butter, soup, and onion in...
1	Awesome Slow Cooker Pot Roast	[2 10.75 ounce cans condensed cream of mushr...	[In a slow cooker, mix cream of mushroom soup....
2	Brown Sugar Meatloaf	[½ cup packed brown sugar, ½ cup ketchup...	[Preheat oven to 350 degrees F (175 degrees C)...

To summarize, the dataset contains 124647 values. The mean title length is 28.285253556042264 characters. The mean ingredient length is 10.565436793504858 ingredients. Lastly, the mean instruction length is 989.6764944202428 characters.

5. Methods

Initially, the GPT-2 model was used, specifically the GPT2Tokenizer and GPT2LMHeadModel. The idea of this GPT-2 recipe generation is that by providing a sequence context string based on food and cooking, the language model will also generate a text related to this, which is often very close to a recipe (von Platen, 2020). This model is based on auto-regression,which means that after a token is produced, it is added to the sequence of inputs. The new sequence then is the input for the next step. It is based on the idea that a word sequence's probability distribution is made up of the product of conditional next word distributions:

$$P(w_{1:T}|W_0) = \prod_{t=1}^{T} P(w_t|w_{1:t-1}, W_0) \text{ ,with } w_{1:0} = \emptyset,$$

W_0 is the initial context word sequence. The sequence's word length T corresponds to the timestep ($t=T$) from which the token is generated. This gives: $P(w_t|w_{1:t-1}, W_0)$.

From this, the decoding methods Sampling and Top-K sampling were used. Sampling randomly chooses the succeeding word w_t based on its conditional probability distribution. $w_t \sim P(w|w_{1:t-1})$. Language generation is not deterministic. The hyperparameter "do_sample" was set to "True". To increase coherence, the latter distribution has to be adapted so that the likelihood of high probability words is increased, and the likelihood of low probability words is decreased. This is done by setting the "temperature" of the softmax to 7.0. Sampling was enhanced by Top-K sampling. This is a technique presented by Fan et al. in 2018. It takes the K most likely following words and redistributes the probability mass among those. We set top_k to 20.

Though proven to be successful, this model is based on fixed general data. We decided to make a more customizable model based on only cooking data, specifically the dataset mentioned in Section 3. This model should recognise the following recipe structure:

```
Ingredients:

Ingredient 1

Ingredient 2

…



Instructions:

A text explaining the instructions
```

In order to train our model towards this structure, we first create a training and validation set. 10% of the data is used for the training set, and 90% for the test set. The small percentage for training is used since otherwise the dataset is simply too large to handle for the model training; and this model already takes hours to train. The training procedure was as follows: The run_lm_finetuning.py script from huggingface was used and a bashscript run_experiments.sh was defined to run the script and to start training the model on the training data in Google Colab. run_lm_finetuning.py contains a number of errors, specifically with importing WarmupLinearSchedule from transformers. Therefore, we define our own WarmupLinearSchedule in the run_lm_finetuning.py and use run_experiments.sh to train the custom model on the recent training data. Lastly, to test whether the model adapts to the wanted format, a test sequence was defined consisting of a list of ingredients from the test set.

6. Results

The initial model GPT-2 model can generate text based on a string of words. Examples of this are shown in **Table 1**. The output length was set to 100 words.

Input	*Output*
Bake 5 eggs. Add them to the tomatoes. Boil the pasta.	*Bake 5 eggs. Add them to the tomatoes. Boil the pasta. Cover for 6 and set out on a large plate and serve as your usual dessert – or, to reheats! – the sauce may work, it's more complex on me. Try a bit too long a way for your tastes or to try the flavours at any level: if you think a bite looks a little boring try just going with your nose, which has your heart so set up you will find that I find*

Table 1 *Examples of the GPT-2 model*

In this example it is clear that the pretrained model was trained on cooking themed texts, but the output does not make a lot of logical sense. The grammar of the sentences is not right and perhaps the model has too little to work with since it does not mention any of the input ingredients again. However it is still clearly a text about cooking and ingredients. What the custom model can not do is take certain ingredients as an input and output a recipe with these ingredients. That is why a new model was made, which is able to perform this task. This model was tested by having certain ingredients as inputs and testing the quality of the generated text. This could be whether or not it used all ingredients and whether the text is fluent or not. An example of this is shown in **Table 2.**

Input	*Output*
Ingredients: *1/4 cup oil* *6 medium onions, chopped* *4 bell peppers, chopped* *3 carrots, chopped* *1 cup string beans, broken into pieces* *3/4 cup peas* *6 tomatoes, chopped* *1/2 teaspoon black pepper* *1 teaspoon dried thyme* *4 cups medium grain brown rice, cooked (cold)* *1/2 cup tomato paste*	*Instructions: Cut 2 large onions, in half lengthwise and place them on a baking sheet. Cover with a plastic baking sheet. Let sit 30 minutes. Set aside. In a bowl, toss in the onion mixture. Add the rice and cook gently. Remove from heat, and then place rice on a lightly oiled baking sheet. Bake at 425 degrees for 10 minutes, or until the top is golden brown. Remove from heat, and then place on serving platter.* *To serve, garnish the cooked rice with chopped parsley and chopped coriander.* *You can serve this appetizer on it's own or with other rice.* *Enjoy!* *Print Author: Baked Rice Salad Print Recipe Prep Time 20 mins Cook Time 20 mins Total Time 24 mins Author: Baked Rice Salad Print Recipe Print Recipe*

Table 2 *Example of the custom model*

The model generates the text based on the ingredients in a very original way. An interesting thing to notice in the tests is that the format of the generated text is quite similar to a normal recipe. It gives instructions on what to do with each ingredient. However it often uses the ingredients in an unusual manner. An example of this can be found in **Table 2** where rice is placed on a lightly oiled baking sheet and baked in the oven. The generated recipes often had an element of baking present. This could be the case, because some pastry recipes were used in the training of the model. This model performs better than the initial model since its sentences are better constructed and structured. Also it uses most of the ingredients that were given to it. Additionally, some recipes include sentences such as “contact me if you have any questions”. This shows the success of implementing a more personal, blogging style of the training data set onto the output.

7. Conclusion

This paper has analyzed two methods of recipe generation. Firstly, a GTP-2 model was used, secondly, a custom model was created based on the (top-k) sampling of transformers. The latter model produced more coherent, well structured and natural sentences than the former. Nevertheless, improvement is required in terms of training set selection, as there seems to be a prevalence of baking instructions, resulting in some unusual steps being suggested. Overall, the custom model is a promising method for quick recipe generation, which adapts exactly to the present ingredients, making food processing very efficient. For future research, it would be interesting to try the model in different languages and possibly adapt to given tools. The model could also be applied to any other area, where a given set of materials needs to be processed.

8. Statement of Contribution

Tim: Ideating and working on custom model
Sergio: Working on the GTP-2 model, assisting on custom model
Emilia: Writing the report, experimenting with models

9. Sources

Ahn, Y. Y., Ahnert, S. E., Bagrow, J. P., & Barabási, A. L. (2011). Flavor network and the principles of food pairing. *Scientific reports*, *1*(1), 1-7.

Garrido-Merchán, E. C., & Albarca-Molina, A. (2018, November). Suggesting cooking recipes through simulation and bayesian optimization. In *International Conference on Intelligent Data Engineering and Automated Learning* (pp. 277-284). Springer, Cham.

Jain, A., & Bagler, G. (2015). Spices form the basis of food pairing in Indian cuisine. *arXiv preprint arXiv:1502.03815*.

Jermsurawong, J., & Habash, N. (2015, September). Predicting the structure of cooking recipes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 781-786).

Lee, Ryan. (2017). Recipe Box. *Eight Portions*, eightportions.com/datasets/Recipes/#fn:1.

Mori, S., Sasada, T., Yamakata, Y., & Yoshino, K. (2012). A machine learning approach to recipe text processing. In *Proceedings of the 1st Cooking with Computer Workshop* (pp. 29-34).

Nedovic, V. (2013, August). Learning recipe ingredient space using generative probabilistic models. In *Proceedings of Cooking with Computers Workshop (CwC)* (Vol. 1, pp. 13-18).

Tasse, D., & Smith, N. A. (2008). SOUR CREAM: Toward semantic processing of recipes. *Carnegie Mellon University, Pittsburgh, Tech. Rep. CMU-LTI-08-005*.

von Platen, P. (2020). How to Generate Text: Using Different Decoding Methods for Language Generation with Transformers. *Hugging Face – On a Mission to Solve NLP, One Commit at a Time.*, Hugging Face, 18 Mar. 2020, huggingface.co/blog/how-to-generate.