Deep Recipe Generator : DRG

Ufuk Özkul
ufukozkul@hacettepe.edu.tr

Mehmet Ali Kandilcik
kandilcik@hacettepe.edu.tr

Abstract

Huge-scale pre-training on wide domain information followed by adaptation to specific tasks or areas is a significant concept in the NLP field. Complete fine-tuning, which updates the entire model parameters, becomes less possible as we build bigger models. Taking large language models that have billions of parameters as an example, it is exorbitantly costly to set up multiple copies of customized models, each having billions of parameters. That is the reason Lora stands for Low-rank adaptation, involved in re-training the LLM phase. It does not calculate the gradient update for the weights of (freezes) the already trained model and infuses factorization matrices which are trainable, by using singular value decomposition into every single layer of the Transformer architectonics which significantly lowers the amount of computational power needed for training the model parameter. In this paper, to generate more reasonable and acceptable food recipes as a downstream task according to the given ingredients, the LoRA approach will be implemented in the Llama 2-7B model which generates text based on the input prompt, to specialize that model on recipe generation downstream task.

1 Introduction

This paper offers a deep learning-based algorithm for reducing food waste by generating recipes from discarded foods. People generally contribute to food waste by discarding ingredients Due to a lack of culinary skill level, zero or less idea about how to efficiently use the discarded foods. Thus, our motivation is according to discarded foods, people can generate recipes with informative and coherent sub-features.

Using the pre-trained model's weights rather than random initialization generally gives better results and is the preferred way because in this approach there is no need for training the model from nothing to an accurate model. The training phase for LLMs often takes days or weeks even with the powerful GPUs. For skipping this long-duration training phase various NLP tasks depend on adapting a single huge-scale, base language model to various downstream tasks. In this paper's case, adapting the Llama 2-7B as a pre-trained LLM to food recipe generation task and for fine-tuning we are using iFOOD recipe datasets over 230K[16]. Such adaptation is often accomplished by finely tuned training, thereby modifying all of the variables of the already trained model. The main disadvantage of finely adjusted training is that the updated version of the model has the same number of parameters as the base model. Moreover, training the updated version of the model needs the same or more computational power than training the base model where LLM is trained a few times a year. In other words, the LLM implementation and fine-tuning provide substantial computational cost and memory efficiency problems. That is the reason LoRA is used in the learning process of LLMs. LoRA is the parameter-efficient training method and it uses matrix decomposition for lower-rank implementation. This decomposition facilitates an efficient representation by utilizing smaller, manageable matrices. The advantage of this method lies in its potential to significantly reduce storage space and faster training.

CMP784 Deep Learning, Fall 2023, Hacettepe University, Ankara, TURKEY.

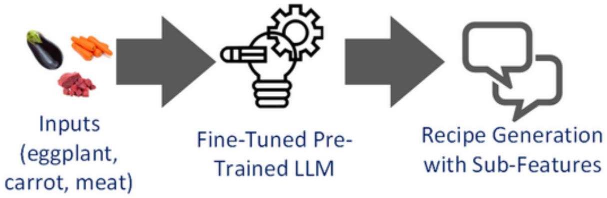

Inputs (eggplant, carrot, meat)
Fine-Tuned Pre-Trained LLM
Recipe Generation with Sub-Features

Figure 1: Figure provides a high-level overview of our deep learning model's procedure. Initially, the user inserts the ingredients of the desired recipe. Following that, it will be evaluated by our deep learning model, which will generate a recipe with sub-features (cooking time, calories, etc.). A well-chosen LLM, a fine-tuning technique, and datasets with optimized hyperparameters are required for this specific method. We are going to discuss our project's research and development procedure.

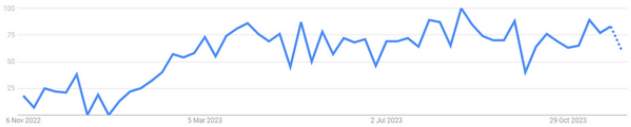


Figure 2: The search popularity graph of the LoRA between 1 November 2022 to 7 December 2023

2 Related Work

Food waste is a problem that has been brought on by an increase in worldwide consumption levels as well as inefficient and occasionally non-utilization of food supplies. A recognized contributing factor to this issue is the general lack of awareness among the public about the proper methods for preparing meals and how to make the most of the ingredients they already have on hand. The literature's current deep-learning-based efforts largely concentrate on classification- and prediction-based techniques for recipe generation to address this problem.

2.1 Dataset

The utilization of multimodal food datasets, which are widely accessible in the literature, is imperative in tackling this broadly recognized problem. Recipe1M+ [14] is one such dataset that is frequently used for recipe generation. Its preference can generally be attributed to two factors: first, the paper 'Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images' provides a detailed explanation of its usage advantages; second, it is the largest publicly available collection of recipe data, encompassing diverse, multimodal data. However, because of its lower computational cost and shorter processing time, the dataset from the 'iFood 2018: Challenge on Fine-Grained Food Classification' [16] competition was used for the project's fine-tuning procedure. A total of 240,888 testing, 10,323 validation, and 101,733 training datasets have been added to the literature by this competition[16].

2.2 Recipe Generation with Fine-Tuned LLM

In the literature, various papers are adopting different methods for recipe generation from ingredients. The paper 'RecipeGPT: Generative Pre-training Based Cooking Recipe Generation and Evaluation System' [10] aims to solve the problem using a fine-tuned GPT-2 model of the Large Language Model (LLM), while 'FIRE: Food Image to REcipe Generation' [9] employs the T5 model for recipe generation. Ingredient extraction and recipe generation are capabilities shared by both papers. Regarding the creation of instructions, or recipes, on test datasets, FIRE, employing a 220M parameter T5 model, scored 25.17 [9], whereas RecipeGPT, using a 355M parameter fine-tuned GPT-2, obtained a Rouge L score of 0.37 [10]. RecipeGPT received an F1 score of 0.77 for ingredient extraction [10], while FIRE received a score of 49.27 [9]. These results clearly show that using fine-tuned, high-parameter LLMs yields higher performance rates. The RecipeGPT research, which also emphasizes the benefits of LLMs with large-scale datasets and the flexibility and advantages of utilizing the same LLM model with varied parameters according to usage limits, supports this finding. Consequently, the project will employ a fine-tuned and proper LLM for its implementation.

2.3 Dataset Format and Procedure of Tokenzier

For our project, one of the issues found is figuring out what format the dataset needs to be transformed into to fine-tune the Large Language Model (LLM). Making sure the LLM performs in a way that supports the project's goals during token prediction is a crucial challenge. A similar problem to ours is addressed in the literature by the Bachelor's thesis 'Cooking Recipes Generator Utilizing a Deep Learning-Based Language Model' [2]. This thesis provides a thorough solution to the topic at hand. The thesis goes into great detail about how the LLM's tokenizer works, how the dataset's parameters (ingredients, instructions, etc.) should be separated to produce a CSV file, and how recipe generation happens after a user enters a desired term. In this thesis, GPT-2 is used as the LLM, and they also provide a comparison with BERT, explaining their choice of GPT-2 [4]. Such an LLM comparison has also inspired the criteria for selecting an appropriate LLM for our project .

2.4 Fine-Tuning Method

The development of language models has led to a significant increase in the number of trainable parameters needed to fine-tune such models, with state-of-the-art models comprising billions of parameters [1]. LoRA is a novel technique that considerably decreases the amount of trainable parameters resulting in lower memory needs and higher training speed. To lower the computational cost with fewer parameters for re-training the LLMs [3]. The 2018 iFood challenge which is based on the recipe generation from cooked meal images increased the papers in the literature about recipe generation such as Inverse cooking[5], which uses Transformer architecture and one hot encoding to train the model but it didn't use the LoRA or any intrinsic dimension technic.

This paper's approach becomes distinct in using transfer learning, LoRA, and intrinsic dimension techniques to the LLM for food recipe generation downstream tasks. In other words, we try to integrate LoRA into LLM for the food recipe generation and increase its quality of it.

3 The Approach

In this title, we will present our proposed mode which is DRG with our research and states. The Creation of the DRG schematic is shown in Figure 3. Our research implementations for developments can be followed in figure 3 with each icon.

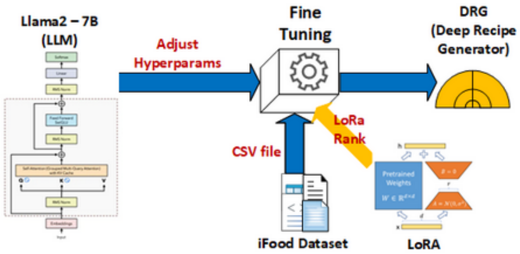


Figure 3: The Creation Process of DRG

3.1 Large Language Model

Large Language Models (LLMs) and word embedding models have been used for many different tasks. Word embeddings hold value for specific tasks and do not need fine-tuning methods whereas they are resource-limited environments, not user-friendly, and have narrower task competency [6]. We will use specific datasets using specific fine-tuning methods on LLMs for training. Large Language Models have specific benefits and drawbacks depending on where they are used.

Our investigation concentrated on choosing available, open-source LLMs for a thorough comparison. After profounded and detailed comparison between LLMs, we chose Llama2-7B for DRG . Despite its relatively lower accuracy than other popular LLMs, such as Mistral-7B and GPT3, it's more efficient and faster than its counterparts, making it the best option for our project [11].

3.2 Why didn't return with Llama2-70B

Before we chose Llama2-70b instead of Llama2-7b because Llama2-70b overperformed to Llama2-7b in accuracy benchmark scores. But while using Llama2-70b we encountered an "Out of memory" error because Llama2-70b files capacity is 137.98GB, while Llama2-7b is 40.43 GB. In the end, we did not have an "Out of memory" error, while using Llama2-7b, so we chose Llama2-7b LLM for our project. Llama2, an open-source LLM, stands out with its pre-training over extensive data and faster inference due to its optimized architecture. Llama2 utilizes a balanced approach to grouped query attention for efficient self-attention.

Llama2 offers a variety of quicker and more effective options [13]. Layer normalization is typically applied after each layer in a transformer block for instance [15]. Llama, on the other hand, substitutes a variant known as Root Mean Square Layer Normalization (RMSNorm), a more straightforward form of layer normalization that has been demonstrated to increase training stability and generalization.

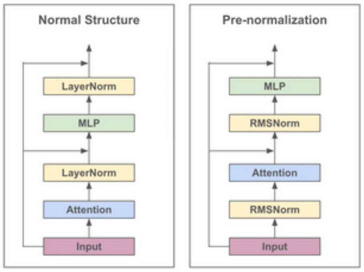


Figure 4: Llama models using RMSNorm instead of LayerNorm

LLaMA models use the SwiGLU activation function in their feed-forward layers, instead of the typical ReLU function used by most neural networks. Although SwiGLU requires more computing power than a standard activation function like ReLU, it has been discovered to produce better results than other activation functions.

$$SwiGLU(x) = swish(xW) \,.\, xV$$
$$swish(x) = x \,.\, sigmoid(\beta x)$$

Figure 5: Formulation of the activation functions

Furthermore, for effective self-attention, Llama2-7b applies a balanced method to grouped query attention. For our project, by appropriately Llama2-7b can be fine-tuned on specific datasets.

3.2.1 Fine-Tuning Method

One of the most recognized methods of fine-tuning, Full fine-tuning prioritizes high accuracy rates, resulting in the update of all parameters of the LLM using specific data during the fine-tuning process [7]. However, as mentioned, attempting to train billions of parameters necessitates great hardware usage and can lead to drawbacks such as intensive demands on time. As a result of our research, we have opted to employ the LoRA method, which is more parameter-efficient for fine-tuning, to adapt the pre-trained LLM for our specific dataset in our project. LoRA (Low-Rank Adaptation) allows to training of some dense layers in a neural network indirectly by optimizing rank decomposition matrices of the dense layers' change during adaptation instead, while keeping the pre-trained weights frozen and offering intrinsic dimensionality. By using intrinsic dimensionality, it finds the redundant rows in the weight matrix which is theoretically a rank-deficient matrix, and removes them. Thus, we have faster training speed, significantly reduce trainable parameters by using LoRA and it make training more efficient and lowers the hardware usage.

To understand "matrix decomposition", simple matrix multiplication in figure 6 was created. In this figure, The overall number of parameters under consideration is significantly diminished. The reduction in the number of parameters that need to be handled simultaneously is substantial and beneficial for computational efficiency. This "r" value, is important because it is a hyperparameter of LoRA and occurs as a trade-off between computational cost and lower parameters.

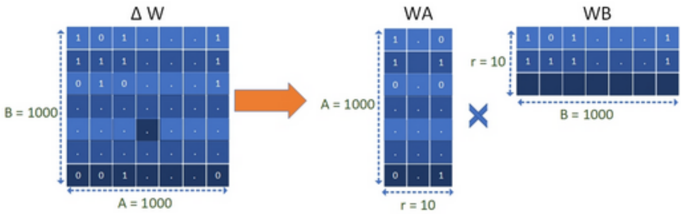
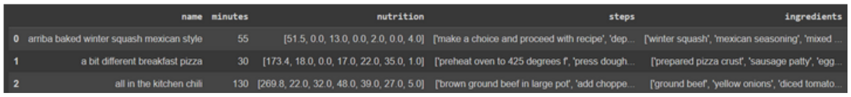


Figure 6: Comparison between word embeddings and LLMs

4 Experimental Results

4.1 Data & Evaluation

The Food.com Recipes and Interactions dataset has 230K unique recipes consisting of recipe names, preparation minutes, nutrition values, preparation steps, number of preparation steps, number of ingredients, and lastly ingredients. According to the replicate library documentation, all of the training data should be included in a single JSON file which is a text-based, file format for preserving structured data. In addition to that the JSON file format should be "prompt": "...", "completion": "..." where in this paper's case the prompt must be 'I have ing1, ing2, ing3, etc. Generate a recipe from those.' completion must be the recipe column of the dataset.



	name	minutes	nutrition	steps	ingredients
0	arriba baked winter squash mexican style	55	[51.5, 0.0, 13.0, 0.0, 2.0, 0.0, 4.0]	['make a choice and proceed with recipe', 'dep...	['winter squash', 'mexican seasoning', 'mixed ...
1	a bit different breakfast pizza	30	[173.4, 18.0, 0.0, 17.0, 22.0, 35.0, 1.0]	['preheat oven to 425 degrees f', 'press dough...	['prepared pizza crust', 'sausage patty', 'egg...
2	all in the kitchen chili	130	[269.8, 22.0, 32.0, 48.0, 39.0, 27.0, 5.0]	['brown ground beef in large pot', 'add choppe...	['ground beef', 'yellow onions', 'diced tomato...

Figure 7: Food 2018 Dataset

After formatting the data into a valid input text for the Llama 2 model, the text will seem like this.

<s>[INST] I have whole berry cranberry sauce, sour cream, prepared horseradish. Generate a recipe from those. [/INST] Recipe Name: berry good sandwich spread. It needs 5 minutes to prepare and it has 79.2 calories, 3.0 total fat, 58.0 sugar, it has no sodium, it has no protein, 6.0 saturated fat. in medium size bowl, coarsely mash cranberry sauce, stir in sour cream and horseradish until well blended. </s>

There are multiple metrics for evaluating the LLM output such as perplexity, human evaluators, BLEU (Bilingual Evaluation Understudy), and ROUGE (Recall-Oriented Understudy for Gissing Evaluation). Furthermore, in the literature, some ground truth data was generated based on the output of GPT-4 and they used it to compare with their model's output. However, there is a limitation with the usage of GPT-4 where asking questions is restricted in a few hours duration. Also, the number of questions asked for GPT-4 is limited. Due to that fact, we will not use GPT-4 for creating the ground truth of the recipes since the Recipes and Interactions dataset already satisfies the real recipes. In short explanations of those evaluation metrics are; perplexity measures of how effectively the model guesses a piece of text.

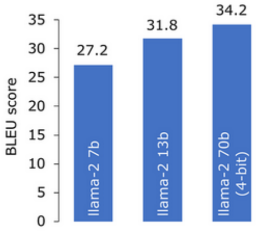


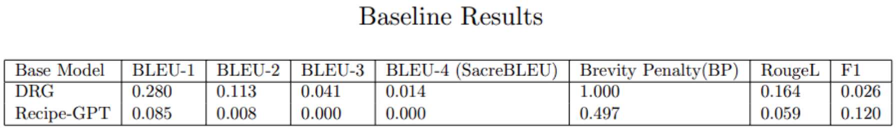
Figure 8: BLUE scores of Llama 2 models

Lower confusion levels suggest improved performance and human evaluators determine the quality of the model's results. They assign ratings to the created replies based on coherence, fluency, relevance, and total grade however this process needs lots of human power which is not suitable for a 230K unique recipe but still this process can be applied to a small portion of the result. BLEU examines the produced result to several reference versions and calculates the degree of similarity. Ratings on the BLEU scale vary between 0 to 1, with higher ratings indicating improved performance. ROUGE computes recall and precision by comparing the produced summary, just like the BLEU, to several reference statements. Also, the duration and computational power needed for generating the result is important which was mentioned in the LLMs Comparison and Choose Reasons section.

4.2 Baseline

We cannot put the results in literature directly since their and our dataset, even the training environment is different, and comparing their results with our results will be a fair comparison. For that case we try to adapt the GPT 2 model to our dataset and train it however the LoRA approach will not work on it and alternatively fully fine-tuning will not be correct in case of fair comparison between the LoRA approach and the fully fine-tuning.

Also, we take the base models from the hugging face site. The Llama 2-7B base model was taken from under the NousResearch username and its model name as Llama-2-7b-chat-hf (NousResearch/Llama-2-7b-chat-hf). Additionally GPT 2 base model was taken from under the bayartsogt username and its model name as mongolian-gpt2 (bayartsogt/mongolian-gpt2).



Baseline Results

Base Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4 (SacreBLEU)	Brevity Penalty(BP)	RougeL	F1
DRG	0.280	0.113	0.041	0.014	1.000	0.164	0.026
Recipe-GPT	0.085	0.008	0.000	0.000	0.497	0.059	0.120

Figure 9: Score comparisons of Deep Recipe Generation DRG and GPT 2 base model

As seen from the above baseline results, the DRG's scores are better than the Recipe-GPT model which uses the GPT 2 model. The main reason behind the score difference is the DRG model uses the Llama 2 model which has 7 Billion parameters. On the other side, the GPT 2 model only uses 124 Million parameters for generating texts. In overall evaluation, the Llama 2 model with 7 billion parameters has better scores and that model has been selected for the parameter efficient training task.

4.3 Main Results

In this paper, we focus only on the training model with various Rank (R) values since training the Llama model which has 7 billion parameters, takes a considerable amount of time and computational power. Due to that fact, we cannot make training with different learning rates and batch sizes. To specify the training environment we set the learning rate 2e-4, Epoch count to 100, batch size to 4, tokenizer to the Llama tokenizer, and multiple rank values such as 2, 8, 16. Since the DRG model is dependent on the Llama model, we cannot make a deep change in the architecture, we have access to only hyperparameters.

As seen from the above results, for better understanding, when we look at the values at epoch 40, we see that the training loss will be decreased when the rank hyperparameter increases which is an expected case because we train the model with the LoRA approach. After all, when the information count is increased, the model can obtain more knowledge and the loss will decrease. Furthermore, we are sure that the model will learn the new data and additional information when we look at to the output of the model. For example when the base model gives this output;

In a blender, combine the milk, vanilla ice cream, and apple juice concentrate. Blend until smooth and creamy. Add the chopped apple to the blender and blend until the apple is fully incorporated and the milkshake is thick and creamy. Pour the milkshake into glasses and serve immediately. You can garnish the milkshakes with additional apple slices or a sprinkle of cinnamon, if desired.

On the other side, the trained model with using 16 ranks gives this output; Recipe Name: apple ice cream. It needs 10 minutes to prepare and it has 145.5 calories, it has no total fat, 10.0 sugar, it has no sodium, it has no protein, it has no saturated fat. combine all ingredients in a blender and blend

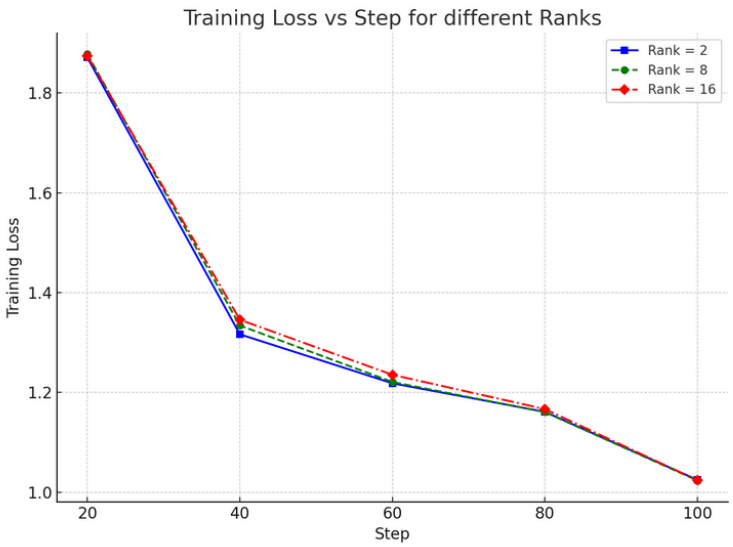
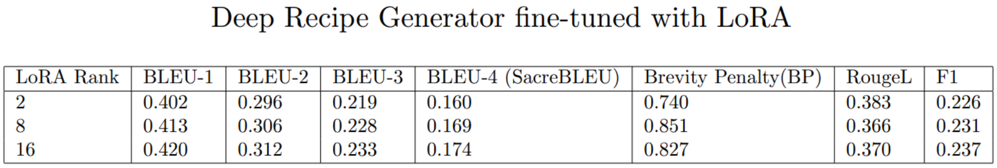


Figure 10: Score comparisons of Deep Recipe Generation DRG and GPT 2 base model

until smooth, pour into a glass and serve immediately. you can add a little more milk if you want a thinner consistency. you can also add a little more vanilla ice cream if you want a thicker consistency. you can also add a little more apple juice if you want a sweeter consistency. you can also add a little more apple if you want a thicker consistency. you can also add a little more vanilla ice cream if you want a thicker.

While analyzing the output, we set the generation length to 250 characters and a trade-off exists. Some recipes are too long and some recipes have fewer steps. When the generation length is set to a low value, the recipes that have many steps, will not predicted correctly. In other words, the generated recipe remains incomplete when compared to the ground truth recipe. When the generation length is set to a high value such as 600 characters, the recipes that have fewer steps, start to repeat the words. For example in the above ice cream recipe output, the model starts to repeat "you can also add a".


Deep Recipe Generator fine-tuned with LoRA

LoRA Rank	BLEU-1	BLEU-2	BLEU-3	BLEU-4 (SacreBLEU)	Brevity Penalty(BP)	RougeL	F1
2	0.402	0.296	0.219	0.160	0.740	0.383	0.226
8	0.413	0.306	0.228	0.169	0.851	0.366	0.231
16	0.420	0.312	0.233	0.174	0.827	0.370	0.237

Figure 11: Trained Models Scores

As seen in above Figure 11, the obtained scores are close to each other. There are no huge differences between the models that have different ranks. When the rank hyperparameter increases the Blue score is also increased but the RougeL score decreases. Similar to the recall and precision case we decided to add the F1 score to decide which model is better than the others. In other words, since the increase in the Blue score ends up with a decrease in the Rouge score and it's not clear which model is past others, we focus on the F1 score for decider purposes. When the rank value increases F1 score is also increased, because the model can receive more information and get knowledge well.

5 Conclusions

To summarise, this work offers LoRA, a unique technique for effectively adapting the Llama 2-7B model for food recipe development, to reduce the disposal of food and generate the recipes with more information such as preparation duration and nutrition values. LoRA reduces trainable parameters to drastically reduce computing costs while finely tuning LLM. The Llama 2-7B was selected as the pre-trained model because of its efficiency, speed, and accuracy. The suggested method has the potential to reduce food waste and generate more informative recipes while also contributing to optimized language models for certain jobs. As a strength of this project, training the model is easy and faster since the model is not fully fine-tuned. The trained additional weights only cover 4MB of additional space whereas the raw LLama 2-7B model covers roundly 13GB of memory. Furthermore, we gain an additional hyperparameter rank (R) for controlling the training and the LoRA approach's additional weights are flexible which allows for adapting the Llama model for various cases. On the other side, the training relies on high-performance GPUs whose its memory must be high to cover all the model's parameters and it takes time to generate recipes (around 1 minute for each recipe). Additionally, our DRG model is dependent on the LLama 2 base model where we cannot have full control such as when we have to use Llama tokenizer and its corresponding input format. Also, a trade-off exists between short and long recipes where we cannot easily decide which one is preferable; repeating in short recipes or incomplete long recipes.

6 Future Work

Add a CNN model for detecting ingredients from the input image then parse the results to our Llama 2 model. In that way, we will create a multi-modal LLM.

7 Acknowledgement

Special thanks to our instructor Erkut Erdem for guidance in the selection of this project's topic and valuable feedback.

References

[1] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. arXiv preprint arXiv:1804.08838, 2018.

[2] Cooking recipes generator utilizing a deep learning-based language model, B. Thesis

[3] Sai Prasanna, Anna Rogers, and Anna Rumshisky. When bert plays the lottery, all tickets are winning. arXiv preprint arXiv:2005.00561, 2020.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805, 2019.

[5] Salvador, A., Drozdzal, M., Giró-I-Nieto, X., & Romero, A. (2018). Inverse Cooking: Recipe Generation from Food Images. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1812.06164

[6] Ruhma K., Demystifying embeddings 101 - The foundation of large language models | Data Science Dojo, August 2022 article blog page

[7] Wu et al., Comparison between parameter-efficient techniques and full fine-tuning: A case study on multilingual news article classification, August 2023, pp. 3-8

[8] Hu et al., LoRA: Low-Rank Adaptation of Large Language Models, June 2021

[9] Prateek Chhikara et al., FIRE: Food Image to REcipe generation

[10] Helena H. Lee et al., RecipeGPT: Generative Pre-training Based Cooking Recipe Generation and Evaluation System

[11] Luzniak K., Is Llama 2 Better Than GPT Models? 6 Main Differences Between Llama 2 vs. GPT-4 vs. GPT-3.5, August 2023, Article at Neoteric

[12] Llama 2 Vs GPT-3.5 Vs GHow Does Llama-2 Compare to GPT-4/3.5 and Other AI

[13] Rozière et al., Code Llama: Open Foundation Models for Code, August 2023, pp. 3

[14] Javier Mar´ın et al., Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images

[15] Cameron R. W., Llama-2 from the Ground Up, August 2023, Newsletter from Rebuy AI company

[16] 'iFood 2018: Challenge on Fine-Grained Food Classification' | Dataset