Natural Language Processing

Recipe Generator Project

Matteo Rigat

Università degli Studi di Milano.

Abstract

This project presents a novel system for generating culinary recipes based on user-provided ingredients. Leveraging the power of pre-trained Large Language Models (LLMs) like GPT-2 and statistical analysis techniques, the system intelligently combines ingredients with suitable cooking methods to generate a recipe. The system is trained on a large dataset of recipes, learning culinary contexts and correlations between ingredients and cooking methods. This knowledge enables the generation of novel recipes that adhere to culinary norms and are likely to be well-received. To generate a recipe the system first use a trained word2vec model to insert in the generation also the cooking methods suitable for the provided ingredients. Generated recipes are finally evaluated against a benchmark of similar real recipes and their user ratings, ensuring the system's output is both creative and of high quality.

Keywords: Recipe Generation, Large Language Models, Fine-tuning, Word2Vec, Recipe Evaluation, Statistical Analysis

1 Introduction

Generating a coherent and culinarily plausible recipe from a predefined, often limited, set of ingredients poses a significant computational challenge. Unlike traditional recipe retrieval systems where users search for a dish and then acquire the necessary components, this project addresses the common scenario of utilizing existing available ingredients, such as those found in a refrigerator, to create a novel set of cooking instructions. This task requires not only linguistic fluency but also an understanding of ingredient compatibility and appropriate cooking procedures.

Recent advancements in Natural Language Processing (NLP), particularly the development of large-scale pre-trained language models, have opened new avenues for tackling complex generative tasks. Foundational to many of these advancements is the Transformer architecture, introduced by Vaswani et al. in their seminal paper "Attention Is All You Need" [1]. By relying entirely on self-attention mechanisms instead of recurrence or convolution, the Transformer enabled models to capture long-range dependencies in text more effectively and efficiently, paving the way for significantly larger and more capable language models. Building upon this architecture, Radford et al. later presented GPT-2 in "Language Models are Unsupervised Multitask Learners" [2], demonstrating that a large Transformer model pre-trained on a massive and diverse text corpus could generate remarkably coherent and contextually relevant text across various domains without task-specific training. Furthermore, GPT-2 showed strong performance when fine-tuned on smaller, domain-specific datasets, making it a prime candidate for specialized generative applications like recipe creation.

This work presents an automated system designed to tackle the ingredient-based recipe generation problem, leveraging the generative power demonstrated by models like GPT-2. The core generative component is the GPT-2 model, which has been specifically fine-tuned on a substantial dataset of recipes from Food.com. This fine-tuning process allows the model to adapt its general language understanding capabilities to the specific structure, vocabulary, and implicit culinary knowledge present in recipes. However, ensuring the generated recipes are culinarily sound requires more than generative fluency alone. To enhance the practical relevance and grounding of the generated output, we integrate a statistical analysis module based on Word2Vec embeddings [3]. This module, trained on the recipe dataset, models semantic relationships between ingredients and cooking techniques, predicting statistically appropriate methods for a given set of ingredients. These predicted techniques, combined with the initial ingredient list, form a structured prompt to guide the fine-tuned GPT-2 model. The LLM then generates the complete recipe steps. Finally, to assess the quality and novelty of the output, we implement an evaluation framework that compares generated recipes against similar real recipes using TF-IDF Cosine Similarity for ingredient matching, and ROUGE-L [5] and BERT-based semantic similarity [4] for comparing the generated steps. This paper details the system's architecture, the dataset preprocessing, the methodology combining Word2Vec and fine-tuned GPT-2, and the evaluation results.

2 Dataset

The success of any machine learning model hinges on the quality and comprehensiveness of the training data. For this project, we utilized a large publicly available recipe dataset from Food.com from Kaggle. This dataset encompasses a vast collection of recipes, each containing detailed information about ingredients, instructions, and user ratings. The dataset is particularly valuable as it provides a real-world representation of culinary diversity, encompassing a wide range of cuisines, cooking styles, and ingredient combinations.

Before feeding the recipe data into the GPT-2 model, we performed several preprocessing steps to ensure data quality and enhance model learning. The preprocessing pipeline included:

- **Data Aggregation:** Recipe content data was merged with user interaction data. For each recipe, the average user rating (avg_rating) and the total number of ratings (rating_count) were calculated from the interactions data and added to the corresponding recipe entry.
- **Quality-Based Filtering:** To focus the model training on high-quality and well-regarded recipes, the merged dataset was filtered. Only recipes meeting specific criteria, namely a minimum number of user ratings (e.g., rating_count $= 2$) and a minimum average rating (e.g., avg_rating $= 4.0$), were retained. Missing values (NaNs) were removed.
- **Technique Extraction via Keyword Matching:** Since the original dataset lacked a dedicated field for cooking techniques, these techniques were identified by analyzing the textual content of the recipe 'steps'. A case-insensitive search was performed within the instructions, matching words against a predefined, comprehensive list of known cooking techniques. The techniques successfully identified through this keyword matching were then stored as a list in a newly created techniques_list column for each recipe.
- **Ingredient Preprocessing:** The preprocessing of the 'ingredients' field involves multiple natural language processing techniques to clean and extract meaningful words. Each ingredient entry is converted to lowercase, and all non-alphabetic characters are removed using regular expressions. Common English stopwords are removed, and very short words are filtered out to eliminate noise. Subsequently, each phrase is tokenized and part-of-speech (POS) tagged. Only nouns are retained based on their POS tags, as they typically carry the core semantic content of an ingredient. For example, both "white sugar" and "sugar" would be normalized to the token `sugar`. These tokens are then lemmatized using WordNet to reduce them to their base forms (e.g., "sugars" to "sugar"). This process is more than redundant when dealing with powerful LLMs. however given the low hardware power available we decided to simplify as much as possible the number of input ingredients during training.

3 Methodology

This project aims to construct a system that addresses the challenge of recipe creation from available ingredients. The goal is to utilize a combination of statistical analysis (Word2Vec) to identify suitable cooking techniques and a fine-tuned Large Language Model (GPT-2) to generate complete, novel recipe instructions based on the input ingredients and predicted techniques. The system utilizes a two-step approach:

3.1 Predicting Suitable Cooking Methods using Word2Vec

Word2Vec is a powerful statistical technique that learns vector representations of words based on their co-occurrence within a corpus of text. In our context, Word2Vec was used to learn relationships between ingredients and cooking techniques. By training Word2Vec on the preprocessed recipe dataset, we obtained vector embeddings for each ingredient and cooking technique. These embeddings captured the semantic similarity between words, allowing us to identify cooking methods that are typically associated with specific ingredients or sets of ingredients.

Given a user-provided list of ingredients, we first preprocess the ingredients using the same pipeline employed for the training data. This ensures that the input ingredients are in a format that is consistent with the Word2Vec model's vocabulary. We then retrieve the corresponding Word2Vec embeddings for each ingredient and calculate the average embedding vector. This average vector represents the overall semantic context of the provided ingredients.

Next, we compare the average ingredient embedding vector with the embedding vectors of all cooking techniques in our vocabulary. Cosine similarity was used as a measure of semantic relatedness between the ingredient context and each cooking technique. The cooking methods with the highest cosine similarity scores are deemed the most suitable for the given ingredients. These predicted cooking methods are then incorporated as input for the GPT-2 model during the recipe generation phase.

3.2 Fine-tuning GPT-2 for Recipe Generation

GPT-2 is a powerful generative language model that has demonstrated remarkable ability in various text generation tasks. We fine-tuned GPT-2 on the preprocessed recipe dataset, enabling it to learn the intricacies of culinary language, ingredient combinations, and the structure of recipes. Fine-tuning involved adjusting the model's parameters to optimize its performance on our specific dataset and task.

To facilitate the model's understanding of the different components of a recipe (ingredients, techniques, steps), we introduced special tokens for each component. These special tokens serve as markers, guiding the model to generate text according to the desired recipe format. By incorporating these tokens into the training data, the model learns to differentiate between the different recipe sections and generate text accordingly.

- **[BOS]** – Beginning of the sentence.
- **[INGREDIENTS]** – Lists the ingredients used in the recipe.
- **[TECHNIQUES]** – Cooking methods to be applied.

- **[STEPS]** – Step-by-step instructions.
- **[PAD]** – Pads sequences to fixed length.
- **[EOS]** – End of the sentence.

During the recipe generation phase, the model receives a structured input string containing the user-provided ingredients and the predicted cooking methods from the Word2Vec model. The input string follows a specific format, utilizing the special tokens to delimit the different recipe sections. For example:

```
[BOS] [INGREDIENTS] tomato, onion, garlic [TECHNIQUES] dice, saute
[STEPS] ... [EOS]
```

The model then generates the remaining parts of the recipe, completing the instruction steps based on its learned knowledge of culinary contexts and ingredient combinations. The generated text adheres to the desired format and utilizes culinary terminology relevant to the predicted cooking methods.

4 Evaluation Framework

Evaluating the quality of automatically generated text, particularly in creative domains like recipe generation, is inherently challenging. Objective assessment requires metrics that capture not only linguistic fluency but also semantic coherence, relevance to the input prompt (ingredients and techniques), and domain-specific plausibility (culinary sense). Our framework addresses this by employing a multi-stage automatic evaluation pipeline designed to compare generated recipes against relevant benchmarks from our dataset.

4.1 Automatic Evaluation Pipeline

The core principle of our automatic evaluation is to assess the generated recipe steps against the steps of real recipes that are culinarily similar. Comparing a generated "tomato soup" recipe against a "chocolate cake" recipe yields meaningless scores. Therefore, our pipeline incorporates a crucial pre-filtering step based on ingredient similarity before applying standard textual similarity metrics. The pipeline proceeds as follows:

1. **Reference Recipe Selection via Ingredient Similarity:** We compare the given ingredient list against the ingredient lists of all recipes in our evaluation dataset. The comparison utilizes Term Frequency-Inverse Document Frequency (TF-IDF) vectorization of the ingredient strings, which allows us to weigh less an ingredient that appears too often in the recipes, followed by Cosine Similarity calculation. This process identifies the *top-K* real recipes from the dataset that share the most similar ingredient profiles with the generated one. This pre-filtering ensures that subsequent evaluations compare the generated steps to relevant and meaningful references.
2. **Step Sequence Evaluation Against Filtered References:** Once the top-K most relevant real recipes are identified, we evaluate the quality of the generated recipe's steps. Both the generated steps (extracted using the `[STEPS]` marker)

and the steps from each of the top-K real recipes are compared using the following metrics:

- **ROUGE-L:** We use the Recall-Oriented Understudy for Gisting Evaluation, specifically ROUGE-L, which measures the longest common subsequence between the generated steps and the reference steps. This metric is particularly useful for evaluating instructional text as it captures sequence overlap and fluency, reflecting how well the order and choice of words align with real recipes, without requiring exact n-gram matches.
- **BERT-based Semantic Similarity:** To assess meaning beyond lexical overlap, we calculate the semantic similarity between the generated steps and the reference steps. We employ a pre-trained BERT model (`bert-base-uncased`) to obtain contextual embeddings for the step sequences. An embedding for each sequence is derived by mean-pooling the last hidden state outputs of the BERT model (considering the attention mask). The Cosine Similarity between the embedding of the generated steps and the embedding of the reference steps provides a score reflecting their semantic relatedness. High scores indicate that the generated steps convey a similar meaning to the steps of comparable real recipes.

4.2 Qualitative Assessment Support

Alongside automatic metrics, qualitative inspection is invaluable. The print highlighted utility, which highlights the user-provided input ingredients and techniques within both the generated output and the reference texts, aids manual review. This allows for quick checks on whether the generated recipe plausibly incorporates the specified inputs and helps identify potential coherence issues or hallucinations missed by the automatic metrics.

5 Results and Discussion

The evaluation framework was applied to recipes generated by the fine-tuned GPT-2 model, prompted with various ingredient sets and their corresponding Word2Vec-predicted techniques. The system consistently produced recipes that were structurally coherent, correctly utilizing the specified special tokens like `[INGREDIENTS]`, `[TECHNIQUES]` and `[STEPS]` to delineate sections. Occasional repetition of these markers within the generated text was observed (e.g., the duplication of `[TECHNIQUES]` ... `[STEPS]` visible in the figures), but the evaluation pipeline's step extraction mechanism, designed to read content after the last `[STEPS]` marker, proved robust for this project.

Quantitative results from the automatic pipeline revealed performance variations depending on the nature of the input ingredients and their similarity to recipes in the evaluation dataset:

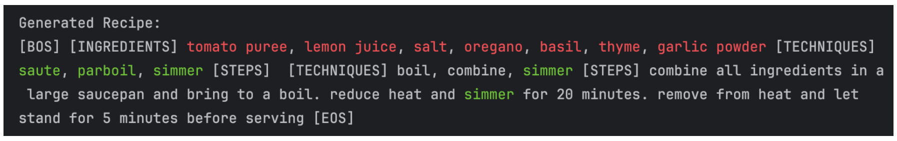
- When generated recipes closely resembled existing high-quality recipes (Example 1, in Figure 1a) with a 0.7474 ingredient similarity, both ROUGE-L score (0.6977) and BERT Similarity score (0.9369) against the top benchmark were notably high. This indicates the model can accurately reproduce known patterns and phrasing when the input aligns well with its training data, generating steps very similar to the real benchmark.

- For more common ingredient combinations that could lead to various dishes (Example 2, pasta ingredients, Figure 1c), the model generated plausible steps but showed more lexical divergence from the top benchmark (ingredient similarity 0.5797). ROUGE-L score was lower (0.2885), reflecting different wording and step choices, but BERT Similarity remained high (0.9135), suggesting the generated steps were semantically coherent and relevant to the ingredients, even if deviating significantly from the specific benchmark recipe structure.
- When presented with highly unusual or nonsensical ingredient combinations (Example 3, including 'ice cream' with 'chicken', Figure 1e), the model demonstrated a tendency towards generating a plausible recipe by selectively ignoring the incongruous input ('ice cream' was omitted from the generated steps). While this attempt at coherence is interesting, it resulted in lower ingredient similarity to the automatically selected benchmark (0.5188) and low ROUGE-L (0.3077). However, BERT Similarity remained remarkably high (0.9447), indicating the generated chicken recipe steps were semantically similar to the benchmark chicken recipe steps, despite the divergent (and ignored) input ingredient.

These results highlight that BERT Similarity effectively captures semantic relevance even when lexical overlap (ROUGE-L) is low due to generation novelty or divergence from benchmarks caused by either valid variation or the model prioritizing plausibility over strict input fidelity.

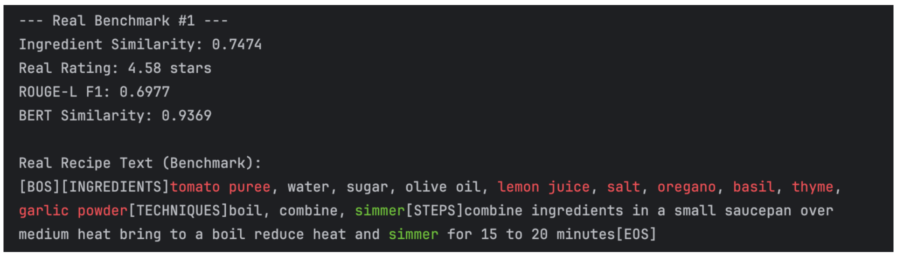
Qualitative assessment using the highlighting tool confirmed that the model generally attempted to incorporate the provided ingredients and techniques within the generated steps. For instance, if 'boil' was provided as a technique, the word 'boil' or related terms often appeared appropriately in the instructions. However, occasional instances of hallucinated ingredients or illogical step ordering were observed. The evaluation strongly confirmed the necessity of the ingredient-based pre-filtering step. Evaluating against randomly selected recipes yielded significantly lower and less meaningful scores, validating the approach of comparing against relevant culinary benchmarks.

Interestingly, the combination of Word2Vec technique prediction and GPT-2 generation sometimes produced recipes perceived as less aligned with the most conventional culinary patterns compared to using the fine-tuned GPT-2 with only ingredients. This may occur because Word2Vec, predicting based on statistical co-occurrence, might suggest techniques that, while related, deviate from the most frequent methods implicitly learned by GPT-2 during fine-tuning. However, this deviation proved valuable for enhancing creativity, prompting the generation of more novel recipes by explicitly introducing techniques that GPT-2 might not have otherwise prioritized based solely on the input ingredients.



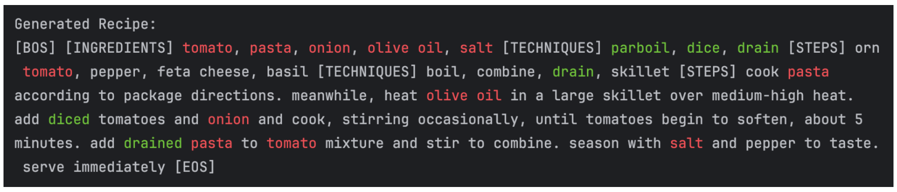
```
Generated Recipe:
[BOS] [INGREDIENTS] tomato puree, lemon juice, salt, oregano, basil, thyme, garlic powder [TECHNIQUES]
saute, parboil, simmer [STEPS]  [TECHNIQUES] boil, combine, simmer [STEPS] combine all ingredients in a
 large saucepan and bring to a boil. reduce heat and simmer for 20 minutes. remove from heat and let
stand for 5 minutes before serving [EOS]
```

(a) Generated recipe from high-similarity input ingredients.



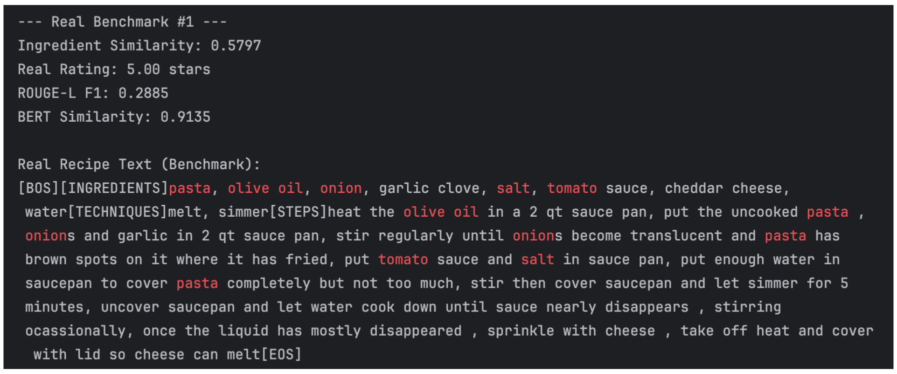
```
--- Real Benchmark #1 ---
Ingredient Similarity: 0.7474
Real Rating: 4.58 stars
ROUGE-L F1: 0.6977
BERT Similarity: 0.9369

Real Recipe Text (Benchmark):
[BOS][INGREDIENTS]tomato puree, water, sugar, olive oil, lemon juice, salt, oregano, basil, thyme,
garlic powder[TECHNIQUES]boil, combine, simmer[STEPS]combine ingredients in a small saucepan over
medium heat bring to a boil reduce heat and simmer for 15 to 20 minutes[EOS]
```

(b) Real benchmark recipe corresponding to the high-similarity input scenario.



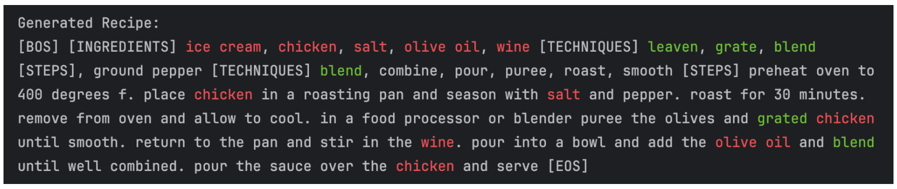
```
Generated Recipe:
[BOS] [INGREDIENTS] tomato, pasta, onion, olive oil, salt [TECHNIQUES] parboil, dice, drain [STEPS] orn
 tomato, pepper, feta cheese, basil [TECHNIQUES] boil, combine, drain, skillet [STEPS] cook pasta
according to package directions. meanwhile, heat olive oil in a large skillet over medium-high heat.
add diced tomatoes and onion and cook, stirring occasionally, until tomatoes begin to soften, about 5
minutes. add drained pasta to tomato mixture and stir to combine. season with salt and pepper to taste.
 serve immediately [EOS]
```

(c) Generated recipe from common input ingredients.



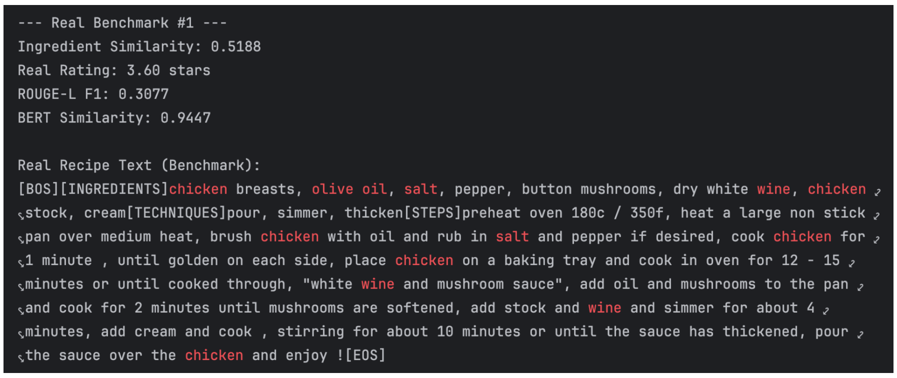
```
--- Real Benchmark #1 ---
Ingredient Similarity: 0.5797
Real Rating: 5.00 stars
ROUGE-L F1: 0.2885
BERT Similarity: 0.9135

Real Recipe Text (Benchmark):
[BOS][INGREDIENTS]pasta, olive oil, onion, garlic clove, salt, tomato sauce, cheddar cheese,
 water[TECHNIQUES]melt, simmer[STEPS]heat the olive oil in a 2 qt sauce pan, put the uncooked pasta ,
 onions and garlic in 2 qt sauce pan, stir regularly until onions become translucent and pasta has
 brown spots on it where it has fried, put tomato sauce and salt in sauce pan, put enough water in
 saucepan to cover pasta completely but not too much, stir then cover saucepan and let simmer for 5
 minutes, uncover saucepan and let water cook down until sauce nearly disappears , stirring
 ocassionally, once the liquid has mostly disappeared , sprinkle with cheese , take off heat and cover
  with lid so cheese can melt[EOS]
```

(d) Real benchmark recipe corresponding to the common ingredients scenario.



```
Generated Recipe:
[BOS] [INGREDIENTS] ice cream, chicken, salt, olive oil, wine [TECHNIQUES] leaven, grate, blend
[STEPS], ground pepper [TECHNIQUES] blend, combine, pour, puree, roast, smooth [STEPS] preheat oven to
400 degrees f. place chicken in a roasting pan and season with salt and pepper. roast for 30 minutes.
remove from oven and allow to cool. in a food processor or blender puree the olives and grated chicken
until smooth. return to the pan and stir in the wine. pour into a bowl and add the olive oil and blend
until well combined. pour the sauce over the chicken and serve [EOS]
```

(e) Generated recipe from unusual/nonsensical input ingredients.



```
--- Real Benchmark #1 ---
Ingredient Similarity: 0.5188
Real Rating: 3.60 stars
ROUGE-L F1: 0.3077
BERT Similarity: 0.9447

Real Recipe Text (Benchmark):
[BOS][INGREDIENTS]chicken breasts, olive oil, salt, pepper, button mushrooms, dry white wine, chicken
stock, cream[TECHNIQUES]pour, simmer, thicken[STEPS]preheat oven 180c / 350f, heat a large non stick
pan over medium heat, brush chicken with oil and rub in salt and pepper if desired, cook chicken for
1 minute , until golden on each side, place chicken on a baking tray and cook in oven for 12 - 15
minutes or until cooked through, "white wine and mushroom sauce", add oil and mushrooms to the pan
and cook for 2 minutes until mushrooms are softened, add stock and wine and simmer for about 4
minutes, add cream and cook , stirring for about 10 minutes or until the sauce has thickened, pour
the sauce over the chicken and enjoy ![EOS]
```

(f) Real benchmark recipe corresponding to the unusual ingredients scenario.

6 Conclusion

This project successfully developed and evaluated a system for generating culinary recipes from a list of ingredients, demonstrating the synergy between statistical methods and large language models. By integrating Word2Vec for predicting relevant cooking techniques and employing a fine-tuned GPT-2 model structured with special tokens, the system proved capable of generating novel yet plausible recipe instructions. The implemented evaluation framework, featuring a critical ingredient-based pre-filtering step followed by ROUGE-L and BERT similarity analysis, provided a meaningful assessment of generation quality against relevant real-world examples. Results indicate the system effectively learns culinary patterns and produces semantically coherent output, while also highlighting current limitations and opportunities for enhancing culinary grounding and fine-grained control.

This work establishes a solid foundation, demonstrating the potential of combining statistical embeddings with LLMs for creative and practical applications in the culinary domain, offering a useful tool for utilizing available ingredients effectively. Looking ahead, the integration of significantly more powerful and recent language models (such as GPT-4 or its successors) coupled with access to more performant hardware could elevate this system beyond a research prototype. With enhanced accuracy, improved handling of complex constraints, and near real-time generation capabilities, a refined version of this framework holds considerable potential for commercialization. Applications could range from sophisticated personalized recipe recommendation apps that minimize food waste to integrated tools for professional kitchen environments.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

[2] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

[3] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781. (Presented at ICLR 2013 Workshop).

[4] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*. (Often cited from NAACL 2019 proceedings as well).

[5] Lin, C. Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In Proceedings of the ACL-04 Workshop on Text Summarization Branches Out (pp. 74-81).