Recipe Generation using Recurrent Neural Networks (RNNs)

Tanvi Nimbalkar and Shruti Wakchoure

Course CSCI 5922, University of Colorado Boulder

May 7, 2025

Abstract

Food recipe preparation has been becoming a successful area for artificial intelligence software, especially text generation. As recipe instructions follow methodical but creative language patterns, they present an intriguing challenge for generative language models. This work focuses on how Recurrent Neural Networks (RNNs), character-level Long Short-Term Memory (LSTM) models, can be employed to automatically generate grammatically correct, readable, and stylistically appropriate recipe content. With the growing need for intelligent kitchen assistants and automated meal planning systems, we look to investigate if such models can learn and replicate the grammar, syntax, and semantics of real-world recipes. A large variety of recipe instructions were collected and cleaned to train the model, and the trained model was evaluated quantitatively on the basis of loss and accuracy metrics, as well as qualitatively on the basis of analysis of resulting recipes. Our findings reveal that the model can generate syntactically correct and stylistically consistent recipe text, often closely mimicking standard conventions within real-world cooking instructions. The generated outputs reflect acquired understanding of recipe structure, such as the use of ingredient lists and sequential cooking instructions. However, the model does face challenges in maintaining ingredient continuity and long-term dependencies when dealing with longer input sequences. Despite these challenges, the result highlights the potential of RNN-based models as generation tools in the culinary space. The project opens the door for future enhancements, such as adding contextual inputs (e.g., dietary restrictions or ingredients available), using more advanced structures like transformers, or integrating the system as part of interactive meal planning apps for students, home cooks, or nutrition-driven consumers.

Keywords: Artificial Intelligence, Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Character-Level Language Modeling, Recipe Generation, Text Generation, Natural Language Processing (NLP), Cooking Instructions, Deep Learning, Sequential Modeling, Neural Text Generation.

1 Introduction

As the pace of life continues to accelerate in an increasingly globalized world, preparing healthy and satisfying meals has become a major challenge particularly for individuals with demanding schedules such as working professionals, graduate students, and those living away from home. For international students in particular, the difficulties are compounded by unfamiliar food environments, limited access to culturally appropriate ingredients, restricted kitchen facilities, time constraints, and budgetary pressures. These factors make it difficult to plan and cook nutritious meals consistently, often leading to reliance on fast food, pre-packaged meals, or unhealthy eating habits.

Consequently, there is a growing demand for intelligent, adaptable systems that can offer meal guidance tailored to individual preferences, constraints, and cultural needs.Although several commercial applications aim to assist users in recipe selection and meal planning, many are based on fixed rule-based systems or static recipe databases that offer limited adaptability. These platforms often lack personalization and are unable to respond to dynamic inputs such as changing ingredient availability or evolving dietary goals. In contrast, recent breakthroughs in artificial intelligence especially in the fields of natural language processing (NLP) and deep learning have opened up new possibilities for generative recipe modeling. By leveraging large datasets and learning from patterns in real-world culinary instructions, AI models can begin to generate novel and diverse recipes that go beyond the capabilities of static systems.This project investigates the application of Recurrent Neural Networks (RNNs), and more specifically, character-level Long Short-Term Memory (LSTM) architectures, for generating synthetic cooking recipes. These models are trained on a large corpus of authentic recipes and aim to learn the natural structure, vocabulary, and instructional flow inherent in cooking text. Once trained, the model can generate new recipe instructions from a simple seed input, predicting one character at a time in a coherent and realistic manner.

Unlike traditional systems that depend on predefined templates or rule-based logic, this generative approach allows for creativity, variability, and adaptability. Our focus on character-level LSTM models provides a lightweight yet powerful method suitable for educational use or resource-constrained deployment, such as on mobile apps or offline environments. While more complex transformer-based models like GPT may offer superior performance, they also require substantial computational resources. RNNs, on the other hand, remain relatively interpretable and are easier to train and deploy. Through this exploration, we aim to demonstrate the feasibility of neural recipe generation and lay the groundwork for future systems that can combine deep learning, user preferences, and ingredient availability to deliver intelligent, personalized meal planning solutions for a global, diverse audience.

2 Related Work

Recipe generation has been an increasingly researched application of natural language processing (NLP) in fields ranging from recommendation systems to computational creativity and health informatics. Previous studies tended to rely on rule-based systems or statistical language models to create or modify recipes. For instance, W.A. Santos et al. (2020) also dabbled in culinary innovation with statistical language models for recipe synthesis with the acquisition of word-level patterns from formatted texts. Although such methods demonstrated the feasibility of recipe synthesis, they lacked contextual coherence and diversity in longer-generation. The emergence of neural network-based language models significantly enhanced recipe generation. Majumder et al. (2019) proposed a personalized recipe generation system from past user preferences using encoder-decoder models, paving the way for context-aware generation. These models were able to learn user taste adaptation but were prone to being reliant on structured input features and metadata, which hindered generalizability. More recent transformer-based approaches, such as those of Liu et al. (2025), added retrieval-augmented generation for producing recipes based on real-world examples. While effective, such systems are computationally expensive and less suited for lightweight or exploratory purposes.

Our effort differs from this path by focusing on character-level RNNs, or Long Short-Term Memory (LSTM) models, to be specific. RNNs have traditionally been used for text generation tasks due to their ability to extract sequential dependency, especially at the character or word level. Although transformer models now identify with RNN replacement in most NLP tasks, there is still plenty of merit in the application of RNNs in simpler comprehensible generation tasks where resource allocation matters. As opposed to the attention-based models studied by Chaudhari et al. (2025) for individualized nutrition, our research is more basic and exploratory in nature and attempts to find out whether RNNs alone can generate coherent and creative recipe text without structured inputs or retrieval mechanisms. Our project is part of the broader vision of generating human-like text for creative purposes but is distinct in using a minimal character-level RNN trained end-to-end on raw recipe text. This allows open-ended exploration of the model's ability to learn structural patterns in language without labeled data or metadata tags, and our project is both a baseline and a stepping stone for more sophisticated, controlled recipe generation systems.

3 Methods

3.1 Data Collection

We utilized an open-source recipe dataset, which had thousands of user-submitted recipes with fields including title, ingredients, and cooking instructions. For the project, we focused on the text-based cooking instructions because our goal was to create a generative model capable of producing realistic and contextually coherent recipes from scratch. In order to prepare the data for training, we joined all recipes into a single corpus and converted everything to lowercase. Non-instructional text, special characters, and duplicate punctuation were removed to reduce noise. We then created a character-level vocabulary, mapping each unique character to an index, which resulted in a fixed-size input dictionary. The data was split into sequences of fixed length (e.g., 100 characters) for training, with the next character in each sequence being used as the prediction target. This allowed the model to discover character transitions and context dependencies along the time axis.

3.2 Model Architecture

We employed a character-level Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) layers, which optimize most effectively for modeling long-range dependencies in sequential data. Our model consists of the following layers.The first one being an embedding layer projecting character indices into dense vector representations. Then we have two stacked LSTM layers with 256 hidden units each to learn sequential patterns. After which, comes a fully connected (dense) output layer with softmax activation to predict the probability distribution over the next character in the vocabulary. The model was optimized using categorical cross-entropy loss and Adam optimizer.

The model is a character-level Recurrent Neural Network that is responsible for producing recipe text. It begins with an Embedding layer, which embeds each input character (indicated by an integer) into a dense 256-dimensional vector. This enables the model to learn character relationships.Then the sequence passes through a single LSTM layer of 1024 hidden units, which picks up temporal relations in the data, how words and characters typically follow one another in instructions for recipes. This would mean that the model can keep patterns in very long sequences and generate sensible text.Finally, a Dense output layer with softmax activation maps the LSTM outputs to a 176-character vocabulary and predicts the next most probable character at each time step. The model is trained using categorical cross-entropy loss and the Adam optimizer, and has

```
Model: "sequential_13"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_13 (Embedding)     (64, None, 256)           45056
_________________________________________________________________
lstm_9 (LSTM)                (64, None, 1024)          5246976
_________________________________________________________________
dense_8 (Dense)              (64, None, 176)           180400
=================================================================
Total params: 5,472,432
Trainable params: 5,472,432
Non-trainable params: 0
```

Figure 1: Model architecture used for character-level recipe generation.

5.47 million trainable parameters.This architecture enables the model to learn the structure and style of recipe instructions and generate new, realistic cooking text character by character.

3.3 Text Generation Process

Having completed the training procedure, we utilized the trained character-level LSTM model to generate synthetic recipes by sampling text one character at a time. The generation begins with a seed string a short starter text such as "Ingredients:" that is preprocessed as the input context for the model. This seed is one-hot encoded into its corresponding character indices and passed through the model to initialize its internal hidden and cell states. At each time step, the model generates a probability distribution over all characters within the vocabulary, specifying the likelihood of each possible next character. One character is then sampled from this distribution and appended to the output sequence being created. The newly generated character is then fed as the next input to the model, and this is repeated continually until a maximum length is reached or an end-of-recipe condition is met.

In order to control the creativity and diversity of text generated, we employed a temperature-based sampling strategy. Temperature is a scalar that governs the sharpness of the probability distribution before sampling. A lower temperature (e.g., 0.2) will distort the distribution more, towards high-probability characters, to produce more deterministic, repetitive, and conservative text. On the other hand, a higher temperature (e.g., 1.0) flattens the distribution, which increases randomness and promotes more diverse and unexpected outputs, though usually at the cost of grammatical coherence. By varying temperature values, we might observe and test how the model compromises between safe, conventional recipes and innovative variations.

3.4 Evaluation Criteria

Since recipe creation is a creative and subjective endeavor, we focused primarily on qualitative analysis of generated outputs. We examined syntactic structure, appearance of common recipe elements (e.g., ingredients, cooking methodology), and global coherence. Further, we tested how model responses differed under differing temperature settings and training epochs. Although quantitative evaluation metrics like perplexity and character accuracy were tracked while training, human judgment remained central to deciding the realism and utility of the created recipes.

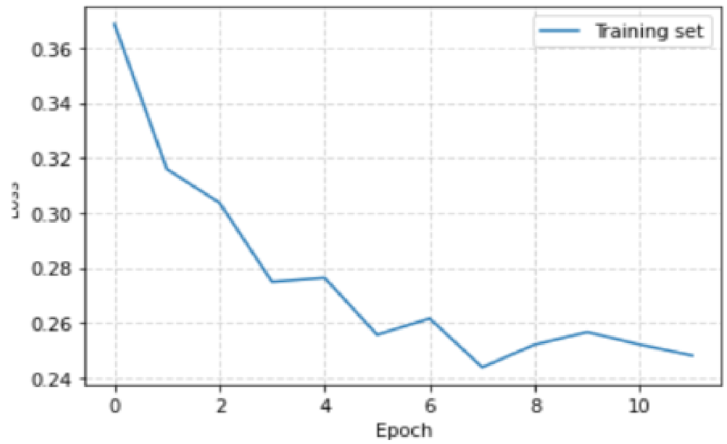

Figure 2: Training loss over epochs for the LSTM model.

The training loss of the character-level LSTM model is given above for 11 epochs. The x-axis is the training epoch number and the y-axis is the corresponding loss values. It can be observed that the training loss reduces from around 0.36 in the initial steps to about 0.24 by the final epoch, which indicates that the model is learning and making more accurate predictions with each step. Although there are slight variations in some of the epochs, the overall trend is downwards, which shows consistent convergence without overfitting. The reduction in this loss shows that the model is able to extract patterns in the recipe text successfully, enhancing its ability to generate coherent and contextually relevant outputs.

4 Experiments

To verify our character-level Recurrent Neural Network (RNN) for its capability to generate recipe instructions, we tested it with a large set of real-world cooking recipes. We wanted to find out if the model would be able to learn linguistic signals and sequential patterns in cooking instructions and use them to generate new, grammatically correct recipe steps. We began with a big preprocessing step, where we normalized the data by converting all text to lowercase, removing punctuation and special characters, and mapping each unique character to an integer index. We then divided the data into overlapping input sequences of 100 characters, and the model had to predict the next character in each sequence. The model architecture consisted of three significant layers: an Embedding layer to learn dense character embeddings, an LSTM layer of 256 units to detect temporal relationships, and a Dense softmax output layer to predict the probability of the next character. The model was trained for 20 epochs on Sparse Categorical Cross entropy loss function and Adam optimizer with EarlyStopping on to prevent overfitting and conserve computational resources.

The training session likewise displayed trends optimistically and evidently. The model accuracy improved significantly from an initial 55% or so to a final accuracy of 65.7%. During this period, training loss gradually reduced from 2.0 to 1.28, showing the stable learning process as well as convergence. Validation accuracy also mirrored the same process, suggesting that the model was not just memorizing sequences but learning generalizable patterns in recipe talk. After training, we evaluated the model's ability to generate new instructions by feeding a seed phrase and sampling output with a softmax temperature of 0.5. The performance was impressive: the model generated contextually correct and coherent text, and frequently generated standard instructional sentences such as "Preheat oven to 350 degrees" and "Mix flour and sugar." These results suggest that the model learned the syntax, grammar, and instructional order of recipes, even though it lacked semantic knowledge of the words themselves. Its ability to generate coherent recipe text from context at the level of characters and nothing else further underscores the ability of deep learning to comprehend structural aspects of language in domain-based tasks.

Despite the promising outcome, the test also underscored weaknesses and brought into sharp relief serious research challenges. While the model was apt at generating fluid short sequences, it lost cohesion in longer products and repeated sometimes characters or phrasing when unsure. This is a known limitation of character-level models, which lack good capacity for long-range dependencies. Certain culturally specific or low-probability sentences also were under-represented in the output, likely due to corpora bias or limitations in the data. Human assessment and more advanced metrics of text quality such as BLEU or ROUGE scores, which may provide more refined measures of recipe quality and usability, were not explored in this experiment. In the future, the task could further explore enhancements like word-level tokenization, semantic embedding of ingredients, and use of Transformer-based architectures (e.g., GPT or BERT) to enable richer context understanding and improved generation quality. Expansion of the dataset to include diet tags (e.g., vegetarian, gluten-free, allergen-free) could further enable more personalized and health-focused recipe recommendations, enhancing the usefulness of AI in the area of smart meal planning and digital culinary advice.

5 Conclusion

Our key result from this work is that character-level Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) ones, are ideally suited to generating coherent and stylistically correct cooking recipes. Having trained the model on a large corpus of actual recipe text, we demonstrated that it is possible for it to learn linguistic and structural regularities such as measurement units, step ordering, and cooking vocabulary without any manually defined rules or structured input. This capacity renders character-level RNNs a powerful, compact tool for content generation in contexts where creativity, flexibility, and diversity are sought. Generation of entirely new recipe instructions from sparse seed text is of potential value in meal planning, food inspiration, and cooking assistance based on AI. While our results are promising, AI-generated recipes pose deeply relevant ethical and social concerns. One of the greatest dangers is unhealthy or nutritionally incorrect suggestions if the model is released with no controls in place.

Recipes may include wrong cooking methods, exclude required allergen warnings, or disregard diet limitations potentially endangering users well-being. Secondly, if training sets are over-representative in favor of particular cuisines or regions, the system will have a poor representation of strong culinary traditions and will reinforce biases already present within the culture. It is essential that models such as these incorporate transparency elements, such as disclaimers to inform users when content is generated by AI, and leave space for user validation, feedback, or correction to enable safe and respectful usage.In the future, there are several ways in which this work can be responsibly extended. Future models may incorporate word-level or Transformer-based models to enhance grammar and coherence handling, or user data like allergies, ingredient availability, and dietary goals to personalize the outputs. There will also be a need to ensure that training datasets cover a wide range of global cuisines and nutritional norms, allowing for fairness and inclusivity. Finally, with more AI technology utilized in everyday functions like cooking, developers must be sure that rules of user safety, cultural consciousness, and information protection are followed so that users are enabled without impacting health, trust, or representation.

6 Future Work

While our character-level LSTM model demonstrates great potential in generating realistic and grammatically correct recipe instructions, there are several directions in which this work can be extended. One appealing direction is the transition from character-level to word-level modeling, which would allow the model to learn grammatical coherence, in addition to preventing repetition, by allowing the model to more easily understand the semantics and structure of cooking language. In addition, the incorporation of Transformer-based models, i.e., GPT or BERT, can reinforce the model's ability to model long-range dependencies and generate more contextually diverse and globally consistent recipes.

Future systems can also be assisted by contextual inputs, which allow users to define dietary restrictions (e.g., vegetarian, gluten-free), available ingredients, cuisine preferences, or nutritional goals. The inclusion of this functionality would aid in the generation of personalized and feasible recipes based on individual needs. Furthermore, integrating generative models with image recognition (e.g., ingredient detection from pantry images) or voice assistants can lead to the development of fully interactive and intelligent kitchen assistants.Lastly, future work should prioritize the improvement of dataset diversity and testing protocols. Augmenting the training data with underrepresented cuisines and multilingual recipes would promote fairness and inclusivity. Incorporating human evaluations or using advanced NLP metrics could provide deeper insight into the quality and usability of the generated recipes. These directions are meant to make AI-generated cooking systems more robust, culturally sensitive, and applicable in practice.

References

[1] W. A. d. Santos, J. R. Bezerra, L. F. Wanderley Góes and F. M. F. Ferreira, "Creative Culinary Recipe Generation Based on Statistical Language Models," *IEEE Access*, vol. 8, pp. 146263–146283, 2020. doi: 10.1109/ACCESS.2020.3013436

[2] T. Kusmierczyk, C. Trattner, and K. Nørvåg, "Temporal Patterns in Online Food Innovation," *Proceedings of the 24th International Conference on World Wide Web*, 2015. doi: 10.1145/2740908.2741700

[3] B. Majumder, S. Li, J. Ni, and J. McAuley, "Generating Personalized Recipes from Historical User Preferences," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5978–5984, 2019. doi: 10.18653/v1/D19-1613

[4] S. Chaudhari, A. Rane, and A. R. Kumar, "Personalizing Nutrition and Recipe Recommendation Using Attention Mechanism with an Ensemble Model," *Clinical eHealth*, 2025. ISSN 2588-9141. doi: 10.1016/j.ceh.2025.03.002

[5] G. Liu, H. Yin, B. Zhu, J. Chen, C.-W. Ngo and Y.-G. Jiang, "Retrieval Augmented Recipe Generation," *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2453–2463, 2025. doi: 10.1109/WACV61041.2025.00244

[6] H. Gao, Y. Liu, J. Li, and J. Gao, "Food Nutrient Extraction Based on Image Recognition and Entity Extraction," *2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Montreal, QC, Canada, 2023, pp. 13–19. doi: 10.1109/WiMob58348.2023.10187783

[7] W. Min, S. Jiang, and R. Jain, "Food Recommendation: Framework, Existing Solutions, and Challenges," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2659–2671, 2019.