ReciFine: Finely Annotated Recipe Dataset for Controllable Recipe Generation

Nuhu Ibrahim, Rishi Ravikumar, Robert Stevens and Riza Batista-Navarro
Department of Computer Science, The University of Manchester, United Kingdom
Correspondence: nuhu.ibrahim@manchester.ac.uk

Abstract

We introduce ReciFine, the largest human-evaluated, finely annotated recipe dataset to date, designed to advance controllable and trustworthy recipe generation. Existing resources, such as RecipeNLG, extract food items only from ingredient lists, overlooking entities expressed in instructions, including tools, chef actions, food and tool states, and durations, which are crucial for realistic and context-aware generation. To address this limitation, we extend RecipeNLG with finely annotated extraction of over 97 million entities across ten entity types from 2.2 million recipes. We are the first to explore recipe generation with explicit control over multiple entity types, enabling models to generate recipes conditioned not only on ingredients but also on tools, chef actions, cooking durations, and other contextual factors. Large language models fine-tuned or few-shot prompted with ReciFine extractions consistently outperform those trained on ingredient-list data alone across both automatic and human evaluations. ReciFine establishes a foundation for factual, coherent, structured, controllable recipe generation, and we release a human-annotated benchmark to support future evaluation and model development.

1 Introduction

Large language models (LLMs) such as GPT-4o (Hurst et al., 2024) and LLaMA (Dubey et al., 2024), along with their contemporaries, are increasingly integrated into everyday life. They are widely applied for text summarisation (Zhang et al., 2020), machine translation (Liu et al., 2020), and even literary composition (Gómez-Rodríguez and Williams, 2023). Over the last decade, several studies have reported that state-of-the-art (SOTA) language models were inadequate for open-ended tasks that require creativity, such as story generation (Fan et al., 2018), poetry (Chakrabarty et al., 2021), and even code generation (Austin et al., 2021). These works highlighted that the models often struggled with coherence, expressiveness, or domain-specific reasoning. Yet, within the same decade, the field has undergone a dramatic shift: LLMs are now relied upon for many of these very tasks, with strong results and potential demonstrated in creative writing (Franceschelli and Musolesi, 2025), figurative language understanding (Ichien et al., 2024), and code generation (Dong et al., 2025). This transformation has been contingent upon immense investment in research and engineering effort, particularly the curation of large-scale and high-quality datasets for training and fine-tuning.

An equally essential yet comparatively underexplored application area for LLMs is food and its preparation. In this domain, SOTA LLMs remain inadequate (Mohbat and Zaki, 2024): while they produce fluent text, they struggle to represent the structured, sequential, and context-dependent nature of cooking. Recipe generation, much like other forms of creative text generation, requires not only grammaticality and fluency but also extensive world knowledge, commonsense reasoning, and discourse modelling to ensure coherence. Crucially, it also demands large, finely annotated corpora for supervision. However, existing recipe datasets are typically either small in scale or annotated at only a shallow level of detail, thereby hindering the development of LLMs capable of generating trustworthy and controllable recipes.

For instance, prior works on automatic recipe generation (Lam et al., 2023; Yu et al., 2020; Taneja et al., 2024; Goel et al., 2022; Bień et al., 2020) have typically relied on the ingredients extracted from recipe ingredient lists alone, ignoring the actual ingredients mentioned in the recipe instructions and their quantities, the tools and chef actions required to prepare the food, the state of the food or tool, and other important factors. In practice, however, ingredients mentioned in recipe direc-

Findings of the Association for Computational Linguistics: EACL 2026, pages 4058–4074

tions are often not the same as those listed, creating mismatches that degrade model performance. In addition, training or fine-tuning LLMs for recipe generation by using only ingredients while ignoring other important entity types encourages the model to hallucinate and generate incomplete or nonsensical recipes, thereby limiting their reliability for practical use.

We address this gap by constructing and releasing ReciFine, the largest finely annotated recipe dataset to date, annotated with ten distinct food entity types, including ingredients, tools, food and tool actions, states of tools and food, durations, and quantities, across over 2.2 million recipes. Our approach builds on Named Entity Recognition (NER), framing recipe entity extraction as a sequence tagging task. We begin with the English Recipe Flow Graph (ERFG) corpus (Yamakata et al., 2020), which contains the most detailed annotations for 300 recipes across 10 important entity types. Using the ERFG dataset and the knowledge-augmented and entity type-specific NER training approach in Ibrahim et al. (2026), we trained encoder-based LLMs (BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)) that outperform prior SOTA results for recipe entity extraction. We then apply the best models to the large-scale RecipeNLG (Bień et al., 2020) dataset, consisting of over 2.2 million recipes, thereby producing ReciFine.

This resource enables new downstream applications. First, we demonstrate controllable creative recipe generation using ingredients extracted from recipe directions rather than ingredient lists, showing that models trained on these signals generate recipes that are more coherent and consistent with gold references. Second, we used the full range of entity types to condition generation on combinations of ingredients and tools, ingredients and chef actions, and a combination of all 10 entity types, providing unprecedented control over the generative process. Finally, we show that control of recipe generation with fine-grained annotations also facilitates recipe adaptation: for example, omitting certain tools or actions in the input naturally leads the model to produce alternatives, yielding recipes aligned with specific user contexts. All code, datasets and pretrained model weights are released publicly under CC BY-NC 4.0[1]. To summarise, the contributions of our work are as follows:

- We present **ReciFine**, the largest finely annotated recipe dataset to date, covering ten entity types and providing over 97 million extracted food entities that were manually validated for reliability.
- We extend the ERFG corpus from 300 to 800 human-annotated recipes, showing that additional annotations improve model performance on recipe information extraction.
- We demonstrate the application of the ReciFine dataset by finetuning open-source LLMs (GPT-2, LLaMA, Mistral, Qwen) and few-shot prompting GPT-4.1 for recipe generation across multiple prompt types (see Section 5.2), marking the first exploration of controllable recipe generation that extends beyond ingredients alone.
- We release the first human-annotated evaluation dataset of automatically generated recipes, enabling future benchmarking of recipe generation and adaptation tasks.

2 Related Work

2.1 Recipe Datasets

Recent years have seen the release of multiple large-scale corpora for computational recipe research. Datasets such as Recipe1M (Marin et al., 2018) and RecipeNLG (Bień et al., 2020) contain millions of recipes scraped from online sources. These resources typically include titles, ingredient lists, food images, and cooking instructions, and have been widely used to train neural models for tasks such as recipe retrieval (Hu et al., 2024), classification (Sakib et al., 2025), and generation (Taneja et al., 2024). However, while their scale is an advantage, these corpora lack comprehensive fine-grained annotations or contain none at all. For example, Recipe1M provides no entity type level annotation, and RecipeNLG includes only a single annotated entity type: ingredients extracted from recipe ingredient lists, without any annotation of entities appearing in the instructions.

Alongside these large-scale resources, several smaller yet more richly annotated corpora have been introduced to represent the semantic structure of recipes. The English Recipe Flow Graph (ERFG) corpus (Yamakata et al., 2020), for example, provides the most detailed annotations that

[1] https://github.com/nuhu-ibrahim/ReciFine

span tools, temperatures, actions, and states. However, its coverage is limited to just a few hundred recipes. Other efforts target narrower subsets of entities: Goel et al., 2024, Diwan et al., 2020, and Komariah and Sin, 2022 focus on extracting food names, states, sizes, quantities, and temperatures, while FoodIE (Popovski et al., 2019) limits its scope to food entities only. These corpora are invaluable for modelling the procedural aspects of cooking, but their limited size prevents them from supporting large-scale training of modern language models. To address this, we extended ERFG by annotating an additional 500 recipes sampled from RecipeNLG, yielding a final corpus of 800 annotated recipes for model development.

Taken together, the state of existing recipe datasets reveals a trade-off: large collections are broad in coverage but shallow in annotation, while fine-grained resources capture rich semantics but remain tiny in scale. No existing dataset combines both properties. This gap motivates our introduction of ReciFine, which provides the first large-scale recipe corpus annotated with ten fine-grained entity types and evaluated by human annotators, thereby enabling both robust information extraction and controllable generation at scale.

2.2 Recipe Named Entity Recognition (NER)

While token classification and NER have advanced owing to the emergence of contextualised encoders such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), their use in the food and recipe domain remains limited. Neural models trained on small corpora, such as AK and GK (Diwan et al., 2020), TASTEset (Wróblewska et al., 2022), and the ERFG corpus, have shown that contextualised encoders can capture fine-grained entities. Recent advances in knowledge-augmented (KA) NER also offer promising directions. For example, question answering-style formulations with external knowledge contexts have been used to guide extraction in specialised domains (Ibrahim et al., 2026; Banerjee et al., 2021), showing improvements in robustness and generalisation. Inspired by these approaches, we adopted an entity type-specific (ETS) and Knowledge-Augmented framework for recipe NER. In particular, we train RecipeBERT-KA and RecipeRoBERTa-KA (based on BERT and RoBERTa encoder models, respectively) on the ERFG corpus, enriching the extraction process with curated knowledge contexts for each entity type. This design (discussed in Section 3.2) enables the models to achieve SOTA performance on ERFG. The inclusion of our newly annotated 500 recipes further enhances performance, demonstrating that both ETS and KA training, as well as human annotations, independently improve model generalisation.

2.3 Recipe Generation

Recipe generation is emerging as an active research area in NLP and food computing. Bień et al. (2020) introduced RecipeNLG, a dataset comprising over 2.2 million recipes that improves the Recipe1M+ (Marin et al., 2018) dataset, and demonstrated its use for semi-structured recipe generation conditioned on ingredient lists. Yu et al. (2020) proposed a routing-enforced generative model, which explicitly guides the generation process to consider ingredients and user dietary category constraints. In addition to automatic metrics, they emphasised human evaluation along five dimensions: readability, accuracy, feasibility, creativity, and overall quality. Lam et al. (2023) trained a ViT5 model (Phan et al., 2022) for Vietnamese recipe generation and Taneja et al. (2024) trained RecipeMC, a recipe text generation method using GPT-2 that relies on Monte Carlo Tree Search. However, these methods primarily condition on ingredient lists and do not incorporate other important contextual factors, such as tools, actions by chef/food/tool, states of food/tool, or quantities. As a result, generated recipes often lack procedural grounding and may diverge from real cooking practices.

Most prior work have evaluated recipe generation using automatic text-similarity metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and GLEU (Sellam et al., 2020), which are limited in their ability to capture true recipe quality. An exception is Yu et al. (2020), which incorporated human evaluation but did not release their evaluation set. To date, no standardised human evaluation dataset exists for recipe generation, hindering reproducibility and fair comparisons across models. To support rigorous evaluation, we also release a human-annotated evaluation set of generated recipes, which allows for pairwise comparison against gold references and enables future benchmarking of recipe generation and adaptation tasks. Moreover, by conditioning on multiple entity types, our approach naturally extends to recipe adaptation: omitting or modifying certain tools or actions in the input leads models to generate plausible alter-

natives, producing recipes tailored to specific user contexts.

3 Methodology

3.1 Data Resources

3.1.1 The English Recipe Flow Graph Corpus

The English Recipe Flow Graph (ERFG) corpus (Yamakata et al., 2020), sourced from Allrecipes.com[2], provides detailed semantic annotations for 300 recipes. It defines 10 entity types (see Table 1). This fine-grained schema captures procedural and contextual information beyond ingredients, making ERFG a foundational resource for modelling cooking processes; however, it is limited in scale.

Tag	Name	Definition
F	Food	Edible items, including both raw ingredients and intermediate products
T	Tool	Cooking tools such as *knives, bowls, pans*
D	Duration	Time durations in cooking (*20 minutes*)
Q	Quantity	Quantities associated with ingredients
Ac	Action by chef	Verbs for deliberate cook actions (*bring* in "Bring the mixture to a boil")
Ac2	Discontinuous Ac	Non-contiguous parts of compound chef actions (*to a boil*)
Af	Action by food	Verbs where food is the agent (*melt, boil*)
At	Action by tool	Verbs denoting tool actions (*grind, beat*)
Sf	Food state	Descriptions of food's state (*chopped, soft*)
St	Tool state	Descriptions of tool readiness (*preheated, greased, covered*)

Table 1: Entity types in the English Recipe Flow Graph corpus.

3.1.2 The RecipeNLG Dataset

RecipeNLG (Bień et al., 2020) is, to date, the largest openly available corpus of cooking recipes, containing over 2.2 million recipes collected from online sources. Its size makes it a valuable resource for large-scale model fine-tuning and training; however, the annotations are limited to `food` entities extracted from the structured ingredient lists. Our exploratory analysis in Appendix A reveals that 92% (2,052,740) of the 2.23 million recipes in RecipeNLG contain at least one ingredient listed in their extracted ingredient list but missing from the cooking instructions. Moreover, approximately 9.9 million of their total 18.9 million extracted ingredients are absent from the instructions, underscoring that relying solely on structured ingredient lists, rather than instructions for extraction, is unreliable. Critically, the recipe instructions themselves remain unannotated with other valuable information, such as food, tools, durations, actions, or states, required to enable controlled recipe generation.

[2] https://allrecipes.com

3.2 Recipe Entity Extraction

3.2.1 Knowledge Augmented (KA) Framework

In contrast to traditional NER, which relies only on the surrounding sentence context, we adopted an approach (Ibrahim et al., 2026) that incorporates KA input. Specifically, curated knowledge prompts are prepended to guide the encoder toward recognising the target entity type (see Fig 1 in Appendix B). Formally, let a recipe instruction be represented as a token sequence $x = x_1, x_2, \ldots, x_n$, and let $p^j = p_1^{(j)}, \ldots, p_m^{(j)}$ denote the textual knowledge context associated with an entity type E_j. The input to the encoder is then constructed as:

$$\tilde{x} = \texttt{[CLS]}, p^j, \texttt{[SEP]}, x_1, \ldots, x_n, \texttt{[SEP]} \quad (1)$$

The knowledge contexts used in the experiment, p^j, are drawn from five types: `Entity Type`, which is the class name; `Definitional Sentence`, a dictionary-style description; `Example`, which consists of annotated examples; `Question Prompt`, a prompt explicitly asking the model to recognise entities; and `Combined`, a combination of all four preceding types.

3.2.2 Entity Type-Specific (ETS) Learning

In addition, we adopt an ETS formulation (Ibrahim et al., 2026). Let $x = x_1, \ldots, x_n$ be a token sequence and $\mathcal{E}$ the set of entity types. Unlike conventional multi-class BIO tagging (BIO-$|\mathcal{E}|$), we constrain the model to identify a single entity type $\mathcal{E}_j \in \mathcal{E}$. Each training or inference example is represented as a tuple $(x, p^j, y^{(j)})$, where p^j is the KA context for $\mathcal{E}_j$ and $y^{(j)} = y_1^{(j)}, \ldots, y_n^{(j)}$ is a BIO label sequence for that entity type only (all other tokens are tagged as `O`).

3.2.3 RecipeRoBERTa-KA and RecipeBERT-KA Models

We trained two encoder models, based on BERT-base (Devlin et al., 2019) and RoBERTa-base (Liu et al., 2019), following the ETS and KA framework described in Sections 3.2.2 and 3.2.1 on the 300 recipes from the ERFG corpus. We denote the resulting models as RecipeBERT-KA and

RecipeRoBERTa-KA, which are subsequently applied to the RecipeNLG dataset to extract fine-grained entities (see Section 4.1).

3.2.4 KA Models' Performance on ERFG

We evaluate the trained models, RecipeRoBERTa-KA and RecipeBERT-KA, using precision (P), recall (R), and F1-score (F1). As a baseline, we compare our results against those presented in the ERFG paper by Yamakata et al. (2020). Their system, referred to as BERT-NER, was built on BERT (Devlin et al., 2019) and trained without the KA and ETS framework, jointly predicting all ten entity types based only on internal sentence context. As shown in Table 2, both models significantly outperform the baseline. Notably, RecipeRoBERTa-KA establishes a new SOTA with an F1 of 90.37 on the ERFG test set, exceeding the previously reported 87.60 from BERT-NER.

Model	Precision	Recall	F1
BERT-NER (Baseline)	86.50	88.80	87.60
RecipeBERT-KA	89.93	89.74	89.33
RecipeRoBERTa-KA	**90.05**	**90.70**	**90.37**

Table 2: RecipeRoBERTa-KA and RecipeBERT-KA performance on the English Recipe Flow Graph corpus.

4 ReciFine Dataset

The limitations outlined in Section 3.1.2 underscore the need for a large-scale dataset with fine-grained annotations, which are suitable for controlled recipe generation. To address this gap, we introduce a new resource, ReciFine, which builds on the RecipeNLG corpus but extends it with detailed annotations for ten entity types (see Table 1) and over 97 million extracted entities. Unlike RecipeNLG, which is restricted to `food` entities extracted from ingredient lists, ReciFine provides human-evaluated token-level annotations within recipe instructions, covering foods, tools, durations, quantities, actions, and states.

4.1 Extracting Recipe Entities

To construct ReciFine, we applied our best-performing model, RecipeRoBERTa-KA, to every recipe instruction in the RecipeNLG corpus. Each sentence was processed independently, and predictions were made for all ten entity types defined in the ERFG scheme. This procedure allowed us to identify and extract fine-grained entities directly from the instructions, rather than relying solely on the structured ingredient lists.

The extraction process involved running the model sequentially over the dataset, with each forward pass restricted to a single entity type following the ETS formulation (see Section 3.2.2). This design ensured that all entities in the RecipeNLG recipes' instructions could be captured with high precision, while reducing the risk of confusion between overlapping categories such as food states, tool states, and action types. The outcome is a large-scale, semantically enriched version of RecipeNLG, which we refer to as ReciFine, containing over 97 million annotated entities across more than two million recipes.

4.2 Manual Validation

Although the trained models achieved SOTA results on the ERFG benchmark, it was necessary to validate their performance when applied at scale to RecipeNLG, which contains over 2.2M recipes. To assess generalisation, we conducted human evaluation on a subset of the finely annotated recipes. Specifically, we recruited two annotators with native-level proficiency in English to manually annotate 500 randomly selected recipes from RecipeNLG, following the annotation guidelines defined in the ERFG corpus. This process allowed us to directly compare the automatic predictions of RecipeBERT-KA and RecipeRoBERTa-KA with human-labeled references, thereby measuring the model's reliability on a new dataset distribution. The results of this evaluation are presented in Section 4.2.1, while inter-annotator agreement between the two annotators is reported in Section 4.2.2.

4.2.1 Evaluating Automatic Annotations

To assess the reliability and validity of the **ReciFine** silver-standard annotations, we compared model-generated labels from RecipeBERT-KA and RecipeRoBERTa-KA against human annotations on 500 randomly selected recipes. As shown in Table 3, the models achieved strong agreement with human labels, confirming that the large-scale automatically annotated ReciFine corpus closely aligns with expert annotation quality. This finding mirrors the results observed on the ERFG corpus (Table 2), further supporting the consistency and robustness of ReciFine's finely annotated extractions.

We then incorporated 300 of the human-annotated recipes (60% of the manually annotated ReciFine subset) into the training data, resulting in 600 annotated examples when combined with

Model	Prec.↑	Rec.↑	F1↑
BERT-NER (Baseline)	79.03	79.75	79.39
RecipeBERT-KA	82.71	81.86	82.28
RecipeRoBERTa-KA	**83.78**	**83.24**	**83.51**
RecipeBERT-KA*	**95.56**	95.86	95.71
RecipeRoBERTa-KA*	95.53	**95.91**	**95.72**

Table 3: Model performance on the ReciFine subset. Models marked with * denote versions trained with an additional 300 human-annotated recipes. Arrows (↑) indicate that higher values are better.

ERFG. As shown by the * models in Table 3, this additional supervision improved precision and recall on the remaining 200 recipes (40%) used for testing. These results highlight that even limited human supervision further enhances the alignment of automatic annotations with expert judgements, reinforcing the reliability, scalability, and trustworthiness of ReciFine as a silver-standard dataset for recipe generation and other related downstream tasks.

4.2.2 Inter-annotator Agreement

To assess the reliability of the human annotations, we measured inter-annotator agreement (IAA) using the F1 score. One annotator's labels were treated as the gold standard, and the second annotator's annotations were compared against them. On the 500 randomly selected recipes, the two annotators achieved an F1 of 91.53%. This level of consistency is comparable to that reported by Yamakata et al. (2020) (F1 = 90.50%), demonstrating that the annotation guidelines were clear and consistently applied.

4.3 ReciFine Descriptive Metrics

ReciFine contains over 2.2 million distinct recipes, making it the largest finely annotated recipe corpus to date. Each recipe is annotated with ten entity types following the ERFG scheme, yielding over 97 million entity mentions. Table 7 in Appendix A shows entity frequencies in ReciFine, and lists the most frequent entities. Common ingredients, such as salt and sugar, dominate the food category, while bowls and ovens are the most frequently used tools. Similarly, verbs such as add, mix, and bake are prevalent among chef actions, reflecting both the scale and linguistic diversity of the dataset. Additional analyses, provided in Appendix A, include the distribution of recipe instructions lengths and ingredient counts, a comparison of ingredients found in instructions versus in ingredient lists, and co-occurrence patterns between entity types, offering further insight into the dataset composition.

5 Controllable Recipe Generation with ReciFine

We demonstrate the downstream application of ReciFine for generating controllable and creative recipes. In this section, we formulate the task, describe dataset preparation, and evaluate generation performance under different types of entity-enhanced prompt types (see Section 5.2). We experiment with both few-shot prompting of LLMs and parameter-efficient fine-tuning (PEFT) approaches, comparing their effectiveness in leveraging the structured information provided by ReciFine. Furthermore, we manually evaluate a subset of the automatically generated recipes and release this human-annotated set as a benchmark dataset to support future research in recipe generation and adaptation.

5.1 Task Formulation

We define **controllable recipe generation** as a conditional text generation task, where a model learns to generate a full recipe sequence $R = \{r_1, r_2, \ldots, r_T\}$ given a structured set of extracted entities $C = \{c_1, c_2, \ldots, c_k\}$ obtained from **ReciFine**. Formally, the objective is to maximise the conditional likelihood:

$$P_\theta(R \mid C) = \prod_{t=1}^{T} P_\theta(r_t \mid r_{<t}, C), \qquad (2)$$

where r_t represents the token predicted at time step t and $r_{<t} = \{r_1, \ldots, r_{t-1}\}$ denotes all previously generated tokens. The model is trained to generate each next token r_t conditioned on the preceding tokens $r_{<t}$ and the structured control context C, thereby allowing the generation process to be guided by specific entity information extracted from ReciFine.

5.2 Entity-enhanced Prompt Types (EPT)

To examine how varying levels of structured information influence recipe generation, we experiment with five **entity-enhanced prompt types**:

1. **Ingredients only from Ingredient List (ING-List)**: using ingredients from recipe ingredient list, which was the baseline approach presented in Bień et al. (2020);

2. **Ingredients only from Instructions (ING-Inst)**: using ingredients from recipe instructions;

3. **Ingredients + Tools (ING+Tool)**: conditioning the model on both food entities and associated cooking tools;

4. **Ingredients + Chef Actions (ING+Cook)**: using ingredients and verbs that describe deliberate actions performed by the cook; and

5. **All Entities (ALL)**: conditioning on the full set of ten entity types defined in ReciFine (see Table 1).

The specific prompt templates used for few-shot prompting and PEFT training across the different EPTs are provided in Appendix E.

EPT	Model	BLEU↑	METEOR↑	WER↓	BERTScore↑
ING-List	GPT-2	0.119	0.314	0.914	0.882
	LLaMA-3 8B	0.151	0.352	0.879	0.892
	Mistral 7B	**0.164**	0.364	**0.868**	**0.895**
	Qwen2.5 7B	0.125	0.334	0.905	0.887
	GPT-4.1	0.065	**0.391**	1.818	0.867
ING-Inst	GPT-2	0.159	0.376	0.886	0.896
	LLaMA-3 8B	0.203	0.420	0.838	0.908
	Mistral 7B	**0.219**	**0.432**	**0.823**	**0.910**
	Qwen2.5 7B	0.182	0.408	0.877	0.903
	GPT-4.1	0.071	0.405	1.829	0.870
ING+Tool	GPT-2	0.190	0.420	0.845	0.907
	LLaMA-3 8B	0.239	0.470	0.804	0.918
	Mistral 7B	**0.254**	**0.479**	**0.781**	**0.919**
	Qwen2.5 7B	0.211	0.452	0.857	0.912
	GPT-4.1	0.077	0.420	1.772	0.873
ING+Cook	GPT-2	0.284	0.510	0.714	0.924
	LLaMA-3 8B	0.348	0.564	0.652	0.933
	Mistral 7B	**0.365**	**0.576**	**0.626**	**0.936**
	Qwen2.5 7B	0.305	0.536	0.722	0.927
	GPT-4.1	0.091	0.449	1.707	0.877
ALL	GPT-2	0.283	0.493	0.812	0.911
	LLaMA-3 8B	0.498	0.702	0.552	0.953
	Mistral 7B	**0.514**	**0.712**	**0.527**	**0.955**
	Qwen2.5 7B	0.448	0.674	0.630	0.946
	GPT-4.1	0.132	0.519	1.533	0.888

Table 4: Evaluation results using automatic text-similarity metrics. Reported values are mean scores computed across all ten independent generations. Appendix C provides the mean scores computed across the best and worst scores observed across these runs. Bold values indicate the best-performing model per context type. Arrows (↑/↓) denote whether higher or lower values are better.

5.3 Model Selection

We evaluate controllable recipe generation using both a baseline and several instruction-tuned large language models (LLMs). As our baseline, we adopt **GPT-2**, consistent with the model used in the RecipeNLG framework for ingredient-list generation (Bień et al., 2020). To explore more capable but computationally efficient alternatives, we select instruction-tuned models of moderate size that: (i) understand task–response prompts, (ii) can be efficiently finetuned on one or two NVIDIA A100 (80GB) GPUs via PEFT methods, and (iii) have open weights to support reproducibility.

Specifically, we experiment with LLaMA-3 (8B) Instruct[3], Mistral (7B) Instruct[4], and Qwen2.5 (7B) Instruct[5], all available on Hugging Face[6] through the Unsloth framework[7]. We employ two complementary modelling strategies: (1) *few-shot prompting*, where we prompted the SOTA premium LLM, GPT-4.1[8], with structured ReciFine contexts, and (2) *parameter-efficient fine-tuning (PEFT)*, where the open-weight models are finetuned using LoRA (Hu et al., 2022) on the same structured prompts to achieve controllable generation with minimal additional parameters. This setup enables a direct comparison between instruction-tuned and lightweight fine-tuning on medium-scale LLMs and few-shot prompting of SOTA commercial models, across different entity-enhanced prompt types. Together, these experiments provide a comprehensive evaluation of how structured and finely annotated entities shape controllable and creative recipe generation. Appendix D presents comprehensive details of the model training configurations and evaluation procedures to ensure transparency and facilitate reproducibility.

5.4 Dataset Preparation

We randomly selected 200 thousand recipes for fine-tuning and 200 distinct recipes for evaluation. Each evaluation recipe was tested across ten runs (200×10), resulting in a total of 2,000 evaluation samples to account for randomness and ensure robustness.

5.5 Recipe Generation Evaluation

We evaluate generation quality using both automatic and human assessments.

Automatic Evaluation: We report n-gram overlap metrics (BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005)) and semantic similarity metric (BERTScore (Zhang et al., 2019)) between generated and reference recipes. Structural accuracy is evaluated using the Word Error Rate (WER) metric to measure overall language quality. Two hundred randomly selected test recipes were each generated ten times (to ac-

[3]https://huggingface.co/unsloth/llama-3-8b-Instruct
[4]https://huggingface.co/unsloth/mistral-7b-instruct-v0.3
[5]https://huggingface.co/unsloth/Qwen2.5-7B-Instruct
[6]https://huggingface.co
[7]https://github.com/unslothai/unsloth
[8]https://openai.com/index/gpt-4-1/

count for randomness) by every model, and the outputs were evaluated using standard automatic text-similarity metrics. The evaluation outcomes are presented and discussed in Section 6.

Human Evaluation: Annotators rated each recipe on a **5-point Likert scale**, $s_i \in \{1, 2, 3, 4, 5\}$, where s_i denotes the score assigned to recipe i, with 1 = very poor and 5 = excellent. Ratings were collected across five dimensions: **readability**, **accuracy**, **feasibility**, **creativity**, and **overall quality**. Readability measures fluency and clarity; accuracy captures correctness of ingredients, tools, and actions; feasibility assesses whether the recipe is realistically executable; creativity evaluates novelty and originality; and overall quality reflects the annotator's holistic impression of the recipe. Detailed annotation instructions and illustration of the annotation tool are in Appendix F.1. Fifty randomly selected test recipes were each generated ten times (to account for randomness) using our best-performing model, **Mistral 7B** and manually evaluated. The results of the evaluation of the generated recipes and inter-annotator agreement (IAA) are discussed in Section 6

EPT	Readability↑	Accuracy↑	Feasibility↑	Creativity↑	Overall↑
ING-List	4.59	4.11	4.08	4.00	4.12
ING-Inst	**4.74**	**4.39**	**4.40**	**4.24**	**4.43**
ING+Tool	**4.74**	4.35	4.38	4.21	4.39
ING+Cook	4.66	4.30	4.35	4.17	4.34
ALL	4.64	4.30	4.33	4.13	4.31

Table 5: Average human evaluation scores (1–5 Likert scale) across five dimensions for different entity-enhanced prompt types (EPT). Higher values indicate better performance.

Evaluation Dimension	Soft Agreement↑
Readability	0.881
Accuracy	0.834
Feasibility	0.776
Creativity	0.750
Overall Quality	0.824
Average	0.813

Table 6: Inter-annotator agreement (IAA) measured using soft agreement between two annotators across five evaluation dimensions. Higher values indicate stronger agreement.

6 Results and Discussion

Table 4 presents the automatic evaluation results across five entity-enhanced prompt types (EPT), comparing fine-tuned and few-shot models using standard automatic text-similarity metrics. Across all metrics, the results show a consistent trend: generation quality improves as richer, finely annotated entity combinations from ReciFine are incorporated into the prompts. Models conditioned on all entity types (**ALL**) yield the highest BLEU, METEOR, and BERTScore values, with the lowest Word Error Rate (WER), followed by those conditioned on prompts that combine ingredients with chef actions (**ING+Cook**) or cooking tools (**ING+Tool**) (see Table 14 in Appendix G for sample generated recipes and related contexts). These results confirm that structured entity information leads to more coherent, grounded, and contextually aligned recipe generation. Interestingly, prompts that use ingredient entities from **ReciFine** (**ING-Inst**) consistently outperform those using ingredient entities from **RecipeNLG** (**ING-List**), indicating that food entities extracted from recipe instructions are more semantically aligned with actual recipe processes than those extracted from ingredient lists alone. Among models, **Mistral 7B** consistently achieves the best performance across all EPTs and evaluation metrics, outperforming larger models such as GPT-4.1 and similar competitive open-weight models. We attribute this to Mistral's robust instruction-tuned foundation, which enable stable and coherent long-context generation. Notably, despite the open-source models' smaller parameter sizes ($\leq$8B), they surpass GPT-4.1 across all metrics, demonstrating that targeted fine-tuning on **ReciFine** yields greater domain grounding than few-shot prompting of even SOTA large models. This reinforces the relevance of ReciFine for reliable generation and other downstream tasks. The human evaluation results further validate these findings. Table 5 summarises ratings across five qualitative dimensions: readability, accuracy, feasibility, creativity, and overall quality. The overall mean rating across all human evaluation metrics and entity-enhanced prompt types is 4.35/5, indicating generally strong human-perceived quality. Recipes generated using **ReciFine** prompts achieve higher scores than those generated using RecipeNLG ingredient lists alone. Finally, the entity type–specific (ETS) and knowledge-augmented (KA) training approach achieves state-of-the-art performance on the ERFG corpus and further validates the quality of the **ReciFine** dataset for controllable and trustworthy recipe generation, achieving 95% accuracy on its human-annotated subset.

7 Conclusion

We presented ReciFine, the largest finely annotated recipe dataset to date, enabling structured

and controllable recipe generation. By extending RecipeNLG with over 97 million entities across ten entity types, we show that incorporating fine-grained contextual information, beyond ingredients, improves the coherence and factual accuracy of generated recipes. To our knowledge, this is the first work to explore recipe generation conditioned on multiple entity types, including tools, chef actions, durations, and many others. Models fine-tuned or few-shot-prompted with ReciFine outperform existing approaches across both automatic and human evaluations. We release ReciFine and its human-evaluated benchmark to support future research on controllable text generation and grounded food modelling.

Limitations

In this work, we focus on textual recipe instructions, establishing a strong foundation for structured and controllable recipe generation through the introduction of ReciFine. Our approach demonstrates that fine-grained entity conditioning substantially enhances coherence and factual accuracy in generated recipes. Building on this, future work will extend ReciFine into multimodal domains, integrating visual and auditory signals from cooking videos and images to support grounded recipe generation and agentic learning. We also plan to explore integrating knowledge graphs and reinforcement feedback to allow interactive and adaptive generation aligned with user preferences. Overall, we see ReciFine as the foundation of a broader research direction toward multimodal, context-aware, and user-controllable food artificial intelligence systems, connecting structured understanding, creativity, and trustworthiness in generative models.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program Synthesis With Large Language Models. *arXiv preprint arXiv:2108.07732*.

Pratyay Banerjee, Kuntal Kumar Pal, Murthy Devarakonda, and Chitta Baral. 2021. Biomedical Named Entity Recognition via Knowledge Guidance and Question Answering. *ACM Transactions on Computing for Healthcare*, 2(4):1–24.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation With Improved Correlation With Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

Michał Bień, Michał Gilski, Martyna Maciejewska, Wojciech Taisner, Dawid Wisniewski, and Agnieszka Lawrynowicz. 2020. RecipeNLG: A Cooking Recipes Dataset for Semi-Structured Text Generation. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 22–28.

Tuhin Chakrabarty, Xurui Zhang, Smaranda Muresan, and Nanyun Peng. 2021. MERMAID: Metaphor Generation With Symbolism and Discriminative Decoding. *arXiv preprint arXiv:2103.06779*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Nirav Diwan, Devansh Batra, and Ganesh Bagler. 2020. A Named Entity Based Approach to Model Recipes. In *2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW)*, pages 88–93. IEEE.

Yihong Dong, Xue Jiang, Jiaru Qian, Tian Wang, Kechi Zhang, Zhi Jin, and Ge Li. 2025. A Survey on Code Generation With LLM-Based Agents. *arXiv preprint arXiv:2508.00083*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 Herd of Models. *arXiv e-prints*, pages arXiv–2407.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. *arXiv preprint arXiv:1805.04833*.

Giorgio Franceschelli and Mirco Musolesi. 2025. On the Creativity of Large Language Models. *AI & Society*, 40(5):3785–3795.

Mansi Goel, Ayush Agarwal, Shubham Agrawal, Janak Kapuriya, Akhil Vamshi Konam, Rishabh Gupta, Shrey Rastogi, Ganesh Bagler, et al. 2024. Deep Learning Based Named Entity Recognition Models for Recipes. *arXiv preprint arXiv:2402.17447*.

Mansi Goel, Pallab Chakraborty, Vijay Ponnaganti, Minnet Khan, Sritanaya Tatipamala, Aakanksha Saini, and Ganesh Bagler. 2022. Ratatouille: A Tool for Novel Recipe Generation. In *2022 IEEE 38th International Conference on Data Engineering Workshops (ICDEW)*, pages 107–110. IEEE.

Carlos Gómez-Rodríguez and Paul Williams. 2023. A Confederacy of Models: A Comprehensive Evaluation of LLMs on Creative Writing. *arXiv preprint arXiv:2310.08433*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. *ICLR*, 1(2):3.

Tianyi Hu, Maria Maistro, and Daniel Hershcovich. 2024. Bridging Cultures in the Kitchen: A Framework and Benchmark for Cross-Cultural Recipe Retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1068–1080.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. GPT-4o System Card. *arXiv preprint arXiv:2410.21276*.

Nuhu Ibrahim, Robert Stevens, and Riza Batista-Navarro. 2026. Knowledge Augmentation Enhances Token Classification for Recipe Understanding. In *EACL*.

Nicholas Ichien, Dušan Stamenković, and Keith J Holyoak. 2024. Large Language Model Displays Emergent Ability to Interpret Novel Literary Metaphors. *Metaphor and Symbol*, 39(4):296–309.

Kokoy Siti Komariah and Bong-Kee Sin. 2022. Enhancing Food Ingredient Named-Entity Recognition with Recurrent Network-Based Ensemble (RNE) Model. *Applied Sciences*, 12(20):10310.

Khang Nhut Lam, Y-Nhi Thi Pham, and Jugal Kalita. 2023. Cooking Recipe Generation Based on Ingredients Using ViT5. In *The International Conference on Intelligent Systems & Networks*, pages 34–39. Springer.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2018. Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *arXiv preprint arXiv:1810.06553*.

Fnu Mohbat and Mohammed J Zaki. 2024. LLaVA-Chef: A Multi-Modal Generative Model for Food Recipes. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 1711–1721.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Long Phan, Hieu Tran, Hieu Nguyen, and Trieu H Trinh. 2022. ViT5: Pretrained Text-to-Text Transformer for Vietnamese Language Generation. *arXiv preprint arXiv:2205.06457*.

Gorjan Popovski, Stefan Kochev, Barbara Korousic-Seljak, and Tome Eftimov. 2019. FoodIE: A Rule-based Named-entity Recognition Method for Food Information Extraction. *ICPRAM*, 12:915.

Nazmus Sakib, GM Shahariar, Md Mohsinul Kabir, Md Kamrul Hasan, and Hasan Mahmud. 2025. Towards Automated Recipe Genre Classification Using Semi-Supervised Learning. *PLOS One*, 20(1):e0317697.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. *arXiv preprint arXiv:2004.04696*.

Karan Taneja, Richard Segal, and Richard Goodwin. 2024. Monte Carlo Tree Search for Recipe Generation Using GPT-2. *arXiv preprint arXiv:2401.05199*.

Ania Wróblewska, Agnieszka Kaliska, Maciej Pawłowski, Dawid Wiśniewski, Witold Sosnowski, and Agnieszka Ławrynowicz. 2022. TASTEset–Recipe Dataset and Food Entities Recognition Benchmark. *arXiv preprint arXiv:2204.07775*.

Yoko Yamakata, Shinsuke Mori, and John A Carroll. 2020. English Recipe Flow Graph Corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5187–5194.

Zhiwei Yu, Hongyu Zang, and Xiaojun Wan. 2020. Routing Enforced Generative Model for Recipe Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3797–3806.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training With Extracted Gap-Sentences for Abstractive Summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating Text Generation With BERT. *arXiv preprint arXiv:1904.09675*.

A Exploring the RecipeNLG and ReciFine Datasets

In Table 7, we present the total number of entities for each entity type in ReciFine, along with the top five entities for each type.

Entity	ReciFine	Top Entities
Food (F)	30,199,222	*salt, water, sugar, butter, flour*
Tool (T)	10,384,889	*bowl, oven, pan, saucepan*
Duration (D)	3,700,982	*10 min, 5 min, 30 min*
Quantity (Q)	4,476,287	*remaining, all, 2, half, 1*
Chef Action (Ac)	30,854,542	*add, bake, mix, stir, cook*
Discont. Ac (Ac2)	2,199,530	*together, to taste, to a boil*
Food Action (Af)	3,023,358	*cool, stand, set, combined*
Tool Action (At)	705,504	*comes out, stand, set*
Food State (Sf)	7,331,595	*hot, tender, smooth, browned*
Tool State (St)	5,022,448	*large, medium, small, 350°*

Table 7: Frequencies of entities for each ReciFine entity type, along with their most frequent entities.

We then examine the relationship between ingredients listed in RecipeNLG and their actual occurrence within the recipe instructions. Table 8 summarises the total number of ingredients that appear exclusively in the ingredient lists, those found in both lists and instructions, and the corresponding number of recipes with these patterns.

Statistic	Count	Portion (%)
Recipe-Level Coverage		
Recipes with $\geq$ 1 extracted ingredient missing in directions	2,052,740	92.0
Recipes with all extracted ingredients found in directions	178,402	8.0
Ingredient-Level Totals		
Extracted ingredients found in directions	9,984,641	52.8
Extracted ingredients not in directions	8,936,609	47.2

Table 8: RecipeNLG extracted ingredient coverage in the instructions. Proportions are relative to total recipes (2.23M) or total extracted ingredients (18.9M).

We also analyse the distribution of instructions and ingredients in ReciFine and RecipeNLG. Table 9 presents the mean, median, and mode for the number of instructions and ingredients per recipe in both datasets. The results indicate that recipe instructions frequently contain a greater number of food entities than those explicitly listed in the ingredient lists.

Feature	Mean	Median	Mode
Number of instructions per recipe	6.61	5.00	4
Number of ingredients per recipe	8.73	8.0	7
Food entities extracted from instructions (ReciFine)	10.63	9.00	8
Food entities extracted from ingredient lists (RecipeNLG)	8.48	8.00	7

Table 9: Descriptive statistics of instruction and ingredient counts in RecipeNLG and ReciFine.

Finally, we analyse the top co-occurring entities across the ten entity types in ReciFine and RecipeNLG (see Table 10) to understand their relational structure and contextual dependencies within recipes.

Entity Type	Top 5 Co-occurring Entities (Recipes)
Food (F) [RecipeNLG]	flour + salt, salt + sugar, flour + sugar, butter + salt, eggs + salt
Food (F) [ReciFine]	pepper + salt, salt + water, flour + salt, butter + salt, salt + sugar
Tool (T)	bowl + oven, bowl + heat, heat + saucepan, heat + skillet, oven + pan
Duration (D)	10 min + 5 min, 10 min + 30 min, 15 min + 5 min , 30 min + 5 min, 10 min + 15 min
Quantity (Q)	half + remaining, all + remaining, 1/2 cup + remaining, 1 cup + remaining, 1 tbsp + remaining
Action by Chef (Ac)	add + cook, add + stir, add + bake , add + mix, bake + mix
Discontinuous Action (Ac2)	out + together, to a boil + to taste, to make + together, out + to make, to taste + together
Action by Food (Af)	combined + cool, cool + set, cool + form, cool + stand, cool + cooled
Action by Tool (At)	comes out + stand, another + comes out, comes out + small, comes out + medium, comes out + cool
Food State (Sf)	hot + tender, hot + smooth, browned + tender, smooth + warm, hot + warm
Tool State (St)	large + medium, large + small, medium + small, large + med-high, large + low

Table 10: Top 5 most frequent co-occurring entity pairs per entity type in ReciFine and for the Food entity type in RecipeNLG.

B Illustration of KA and ETS NER Approach Compared to Traditional NER

Figure 1 shows the comparison between traditional multi-entity type named entity recognition (NER) and knowledge-augmented (KA) & entity type-specific (ETS) NER approach. While the traditional model predicts all entity types jointly and without knowledge context, our method predicts only the BIO tags relevant to the given context (e.g., only the green tokens are predicted when the context is about *Action by Chef (Ac)*). Tokens unrelated to the target entity type are assigned the `O` label (not shown).

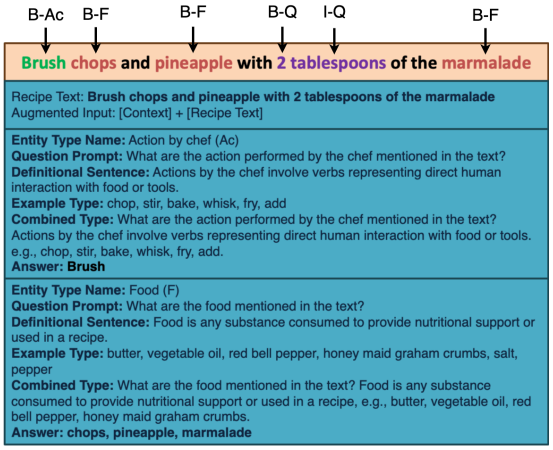

Figure 1: Comparison between traditional multi-entity type NER (top), and KA & ETS NER approach (third and fourth).

C Best and Worst Recipe Generation Evaluation Metrics

To complement the mean of automatic text-similarity metrics over all independent generation reported in Table 4, we present the mean over best (highest) and worst (lowest) performance across ten independent generation runs for each entity-enhanced prompt type (EPT) in Tables 11 and 12, respectively.

EPT	Model	BLEU↑	METEOR↑	WER↓	BERTScore↑
	GPT-2	0.159	0.388	0.832	0.896
	LLaMA-3 8B	0.202	0.432	0.790	0.907
ING-List	**Mistral 7B**	**0.210**	**0.443**	**0.778**	**0.908**
	Qwen2.5 7B	0.172	0.412	0.815	0.901
	GPT-4.1	0.085	0.427	1.600	0.873
	GPT-2	0.206	0.453	0.781	0.909
	LLaMA-3 8B	0.255	0.499	0.727	0.920
ING-Inst	**Mistral 7B**	**0.272**	**0.511**	**0.714**	**0.922**
	Qwen2.5 7B	0.230	0.485	0.765	0.915
	GPT-4.1	0.093	0.446	1.609	0.877
	GPT-2	0.240	0.493	0.747	0.918
	LLaMA-3 8B	0.239	0.543	0.691	0.928
ING+Tool	**Mistral 7B**	**0.307**	**0.554**	**0.674**	**0.930**
	Qwen2.5 7B	0.257	0.524	0.745	0.922
	GPT-4.1	0.098	0.458	1.572	0.880
	GPT-2	0.341	0.588	0.613	0.934
	LLaMA-3 8B	0.348	0.637	0.652	0.943
ING+Cook	**Mistral 7B**	**0.364**	**0.650**	**0.626**	**0.945**
	Qwen2.5 7B	0.305	0.613	0.605	0.937
	GPT-4.1	0.091	0.489	1.707	0.885
	GPT-2	0.348	0.593	0.671	0.914
	LLaMA-3 8B	0.558	0.767	0.431	0.961
ALL	**Mistral 7B**	**0.570**	**0.773**	**0.417**	**0.963**
	Qwen2.5 7B	0.510	0.747	0.490	0.955
	GPT-4.1	0.179	0.570	1.239	0.901

Table 11: Evaluation results using automatic text-similarity metrics on the best of ten generations per model and extracted prompt context type (EPT). Bold values indicate the best-performing model per context type. Arrows (↑/↓) denote whether higher or lower values are better.

EPT	Model	BLEU↑	METEOR↑	WER↓	BERTScore↑
	GPT-2	0.034	0.243	1.048	0.862
	LLaMA-3 8B	0.075	0.275	1.056	0.876
ING-List	**Mistral 7B**	**0.084**	**0.287**	**1.038**	**0.879**
	Qwen2.5 7B	0.056	0.255	1.071	0.871
	GPT-4.1	0.042	0.355	2.143	0.861
	GPT-2	0.064	0.305	1.043	0.883
	LLaMA-3 8B	0.117	0.344	1.038	0.894
ING-Inst	**Mistral 7B**	**0.129**	**0.356**	**1.008**	**0.896**
	Qwen2.5 7B	0.102	0.336	1.034	0.890
	GPT-4.1	0.048	0.369	2.173	0.864
	GPT-2	0.095	0.346	0.977	0.881
	LLaMA-3 8B	0.152	0.399	1.001	0.906
ING+Tool	**Mistral 7B**	**0.160**	**0.404**	**0.917**	**0.908**
	Qwen2.5 7B	0.127	0.380	1.044	0.899
	GPT-4.1	0.054	0.382	2.110	0.867
	GPT-2	0.185	0.433	0.836	0.908
	LLaMA-3 8B	0.249	0.488	0.807	0.923
ING+Cook	**Mistral 7B**	**0.264**	**0.498**	**0.751**	**0.925**
	Qwen2.5 7B	0.203	0.455	0.949	0.915
	GPT-4.1	0.065	0.409	2.043	0.870
	GPT-2	0.173	0.385	0.998	0.761
	LLaMA-3 8B	0.394	0.634	0.741	0.944
ALL	**Mistral 7B**	**0.413**	**0.644**	**0.684**	**0.945**
	Qwen2.5 7B	0.333	0.595	0.854	0.931
	GPT-4.1	0.096	0.468	1.855	0.879

Table 12: Evaluation results using automatic text-similarity metrics on the worst of ten generations per model and extracted prompt context type (EPT). Bold values indicate the best-performing model per context type. Arrows (↑/↓) denote whether higher or lower values are better.

D Models Training and Evaluation Details

D.1 Fine-tuning GPT-2 Baseline

The GPT-2 baseline model was fine-tuned using the Hugging Face `Trainer` API on the filtered RecipeNLG and ReciFine datasets. Tokenisation was performed with truncation and padding to a maximum sequence length of 512 tokens. Training was run for three epochs with the following configuration:

- Batch size: 2 per device (both train and eval)
- Optimiser: AdamW
- Learning rate schedule: linear warmup with 500 steps
- Weight decay: 0.01
- Max sequence length: 512 tokens
- FP16 mixed precision enabled

The fine-tuned model serves as the baseline for comparison with the LoRA-based instruction-tuned models.

D.2 LoRA Fine-tuning of Instruction-tuned Models

Open-source instruction-tuned models (`LLaMA-3 8B`, `Mistral 7B`, and `Qwen2.5 7B`) were fine-tuned using the **Unsloth** framework.[9] LoRA (Hu et al., 2022) parameter-efficient fine-tuning (PEFT) was applied to reduce memory overhead while maintaining strong adaptation performance.

Generation Setup: For evaluation, models were switched to inference mode and generated completions with *Temperature:* 0.7, *Top-p:* 0.9, and *max_new_tokens*: 1024.

D.3 Few-shot Prompting with GPT-4.1

For comparison against proprietary SOTA models, we performed few-shot prompting using GPT-4.1 via the OpenAI API. For each extracted prompt context type (see Section 5.2), a fixed set of $k = 3$ randomly sampled few-shot examples from the training set was prepended to the test prompt. Generation was configured as: *temperature* = 0.7, *top*-p = 0.9, *max_tokens* = 1024.

[9] https://github.com/unslothai/unsloth

D.4 Training RecipeRoBERTa-KA and RecipeBERT-KA Encoder Models

To train the encoder models for fine-grained entity extraction in **ReciFine**, we employed a single NVIDIA A100 GPU (80 GB memory) with an effective batch size of 256 to maximise GPU utilisation and training efficiency. The models were implemented using the `HuggingFace Transformers` library and optimised with AdamW, applying a weight decay of 1×10^{-8} and a learning rate of 2×10^{-5}. Each model was trained for up to 30 epochs.

D.5 Summary of Key Hyperparameters

Table 13 provides a summary of the important training, fine-tuning or few-shot prompting hyperparameters.

Setting	GPT-2	Unsloth Models	GPT-4.1 (Few-shot)
Epochs	3	1	–
Batch size	2	4 (×8 grad. acc.)	–
Learning rate	–	4.2×10^{-5}	–
Max sequence length	512	2048	1024 (generation)
Precision	FP16	BF16/FP16	FP32 (API)
LoRA rank (r)	–	16	–
Temperature	–	0.7	0.7
Top-p	–	0.9	0.9
Max new tokens	–	1024	1024

Table 13: Summary of hyperparameters used for GPT-2 fine-tuning, LoRA fine-tuning, and GPT-4.1 few-shot prompting.

E LLM Prompts for the 5 EPTs

Figures 2–6 present example prompts used for the five entity-enhanced prompt types (see Section 5.2) in fine-tuning the instruction-tuned models or few-shot prompting the premium GPT4.1 model.

Instruction:

You are an expert chef who specialises in writing clear and detailed cooking instructions.
Given the recipe list of ingredients and the key entities involved,
write step-by-step instructions that explain how to prepare the dish.
Your instructions should be accurate, easy to follow, and match the provided input.

Input:

Title: Grilled Turkey & Pineapple
Ingredients from the instruction: loin turkey chops, pineapple, orange marmalade, yogurt, cashew halves
Ingredients from the list:

- 4 top loin turkey chops, boneless, 3/4-inch-thick
- 1 fresh pineapple, peeled and cored
- 3 tablespoons orange marmalade
- 12 cup plain yogurt
- 14 cup cashew halves, roasted and lightly salted, coarsely chopped

Response:

Figure 2: Example prompt for ING-List: using only ingredients from the recipe ingredient list (baseline).

Instruction:

You are an expert chef who specialises in writing clear and detailed cooking instructions.
Given the recipe list of ingredients and the key entities involved,
write step-by-step instructions that explain how to prepare the dish.
Your instructions should be accurate, easy to follow, and match the provided input.

Input:

Title: Grilled Turkey & Pineapple
Ingredients from the instruction: turkey, salt, pepper, pineapple, chops, marmalade, yogurt, nuts
Ingredients from the list:

- 4 top loin turkey chops, boneless, 3/4-inch-thick
- 1 fresh pineapple, peeled and cored
- 3 tablespoons orange marmalade
- 12 cup plain yogurt
- 14 cup cashew halves, roasted and lightly salted, coarsely chopped

Response:

Figure 3: Example prompt for ING-Inst: using ingredients extracted from recipe instructions.

Instruction:

You are an expert chef who specialises in writing clear and detailed cooking instructions.
Given the recipe list of ingredients and the key entities involved,
write step-by-step instructions that explain how to prepare the dish.
Your instructions should be accurate, easy to follow, and match the provided input.

Input:

Title: Grilled Turkey & Pineapple.
Ingredients from the instruction: turkey, salt, pepper, pineapple, chops, marmalade, yogurt, nuts.
Tools from the instruction: charcoal grill, rack, uncovered, grill, thermometer, plates.
Ingredients from the list:

- 4 top loin turkey chops, boneless, 3/4-inch-thick
- 1 fresh pineapple, peeled and cored
- 3 tablespoons orange marmalade
- 12 cup plain yogurt
- 14 cup cashew halves, roasted and lightly salted, coarsely chopped

Response:

Figure 4: Example prompt for ING+Tool: combining ingredients and associated cooking tools.

F Evaluation Tools and Guidelines

F.1 Recipe Generation Human Evaluation Guide and Tool

Figures 9 and 7 illustrate the annotation tool interface and guide for human evaluation of the automatic recipe generation, respectively.

F.2 Recipe Annotation Tool for Food Information Extraction

Figures 8 illustrate the annotation tool interface for food information extraction from recipes.

G Sample Prediction Using Different Entity-enhanced Prompt Types

To qualitatively illustrate the impact of different entity-enhanced prompt types on controllable recipe generation, Table 14 presents sample generations produced under five entity-enhanced prompt types. Each example shows how conditioning the model on progressively richer contextual informa-

Instruction:

You are an expert chef who specialises in writing clear and detailed cooking instructions.
Given the recipe list of ingredients and the key entities involved,
write step-by-step instructions that explain how to prepare the dish.
Your instructions should be accurate, easy to follow, and match the provided input.

Input:

Title: Grilled Turkey & Pineapple.
Ingredients from the instruction: turkey, salt, pepper, pineapple, chops, marmalade, yogurt, nuts.
Chef actions from the instruction: sprinkle, cut, set aside, place, chops, turn, add, brush, grill, inserted, turning, combine, arrange, top, sprinkle, makes.
Ingredients from the list:

- 4 top loin turkey chops, boneless, 3/4-inch-thick
- 1 fresh pineapple, peeled and cored
- 3 tablespoons orange marmalade
- 12 cup plain yogurt
- 14 cup cashew halves, roasted and lightly salted, coarsely chopped

Response:

Figure 5: Example prompt for ING+Cook: combining ingredients and chef actions.

Instruction:

You are an expert chef who specialises in writing clear and detailed cooking instructions.
Given the recipe list of ingredients and the key entities involved,
write step-by-step instructions that explain how to prepare the dish.
Your instructions should be accurate, easy to follow, and match the provided input.

Input:

Title: Grilled Turkey & Pineapple.
Ingredients from the instruction: turkey, salt, pepper, pineapple, chops, marmalade, yogurt, nuts.
Chef actions from the instruction: sprinkle, cut, set aside, place, chops, turn, add, brush, grill, inserted, turning, combine, arrange, top, sprinkle, makes.
Discontinous chef actions from the instruction: None.
Food actions from the instruction: None.
Tool actions from the instruction: None.
Durations from the instruction: 4 minutes, 3 to 5 minutes more.
Tool states from the instruction: medium, 160 degrees.
Food states from the instruction: 1, 4 servings.
Quantities from the instruction: 1 / 2, 2 tablespoons, remaining, 4.
Tools from the instruction: charcoal grill, rack, uncovered, grill, thermometer, plates.
Ingredients from the list:

- 4 top loin turkey chops, boneless, 3/4-inch-thick
- 1 fresh pineapple, peeled and cored
- 3 tablespoons orange marmalade
- 12 cup plain yogurt
- 14 cup cashew halves, roasted and lightly salted, coarsely chopped

Response:

Figure 6: Example prompt for ALL: conditioning on all ten entity types defined in ReciFine.

tion, from ingredients alone to the full set of ten entity types, enhances coherence, specificity, and alignment with real cooking procedures.

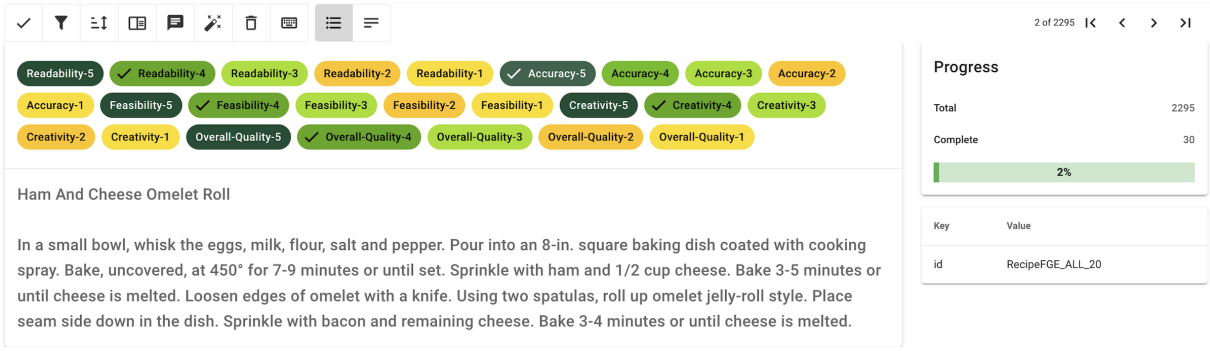

Figure 7: Doccano interface for manually evaluating generated recipes.

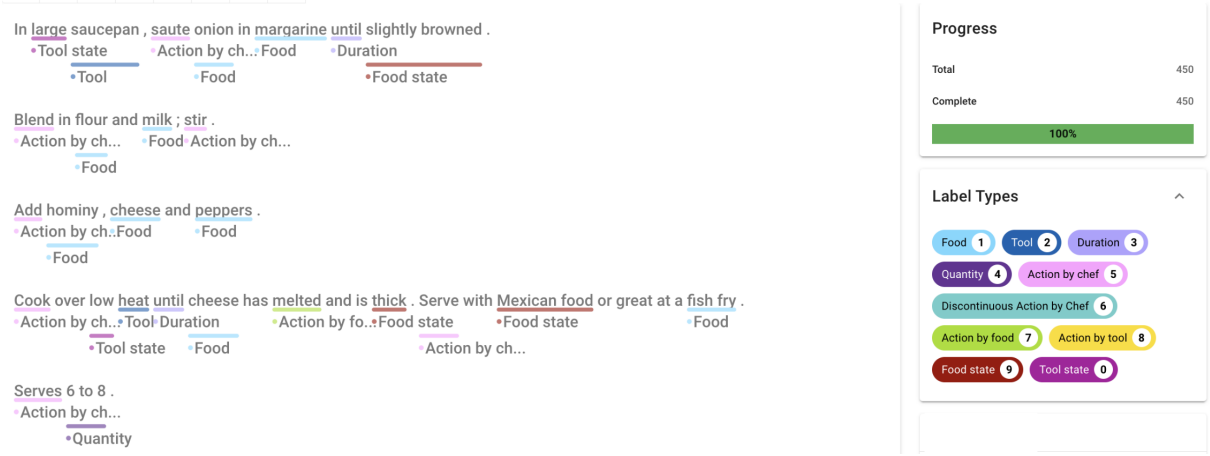

Figure 8: Doccano interface for manually extracting food entities.

Recipe Title: *Caramelized Onion Tart* – (www.epicurious.com/recipes/member/views/caramelized-onion-tart-50166020)	
Food (F in RecipeNLG)	Crust, Flour, Mix all together, Filling, Onions
Food (F in ReciFine)	Butter, Onions, Cream, Balsamic vinegar, Salt, Pepper, Mixture, Crust
Tool (T)	Oven, Skillet, Heat, Baking pan
Duration (D)	30 minutes
Quantity (Q)	1/2 cup
Chef Action (Ac)	Preheat, Add, Melt, Cook, Stir, Allow to thicken, Spread
Discont. Action (Ac2)	To taste
Food Action (Af)	Melted, Browned, Thicken
Tool Action (At)	Melted
Food State (Sf)	Melted, Thickened
Tool State (St)	Hot
Ingredient List	1 1/2 c. All-purpose flour 1/4 Teaspoon salt 7 Tablespoons unsalted butter 4 Tablespoons ice cold water (more if needed) 6 Medium-sized onions Sliced 1/2 cup unsalted butter 1/2 Heavy cream Mix all together 2 Tablespoons balsamic vinegar Salt
ING-List	Preheat oven to 400 degrees. Saute onions in butter over medium-high heat until browned. Add cream, vinegar, salt, pepper, and thyme. Continue to saute until liquid is evaporated. Spread onion mixture into tart shell. Bake for 30 minutes. Cool before serving.
ING-Inst	Melt butter in skillet and saute onions until golden brown. Add cream and balsamic vinegar. Season with salt & pepper. Cook until mixture is thickened. Pour into crust. Bake at 400 degrees for about 30 minutes or until crust is golden brown.
ING+Tool	Preheat oven to 375°. Melt butter in large skillet over medium heat. Add onions and cook, stirring occasionally, until golden brown, about 30 minutes. Add cream and balsamic vinegar. Season with salt and pepper. Cook until mixture is reduced to a syrupy consistency, about 5 minutes. Spread mixture evenly in crust. Bake until crust is golden brown, about 30 minutes.
ING+Cook	Preheat oven to 350 degrees. In a large skillet, add butter and melt over medium-high heat. Once melted, add onions and cook, stirring occasionally, until browned. Stir in cream and balsamic vinegar. Add salt and pepper to taste and allow to thicken for about 10 minutes. Spread mixture over crust and bake for 30-40 minutes or until golden brown.
ALL Entities	Preheat oven to 350F. In a large skillet, add butter and melt over medium heat. When melted, add onions and cook, stirring occasionally, until browned. Stir in cream, balsamic vinegar and salt & pepper to taste, and allow to thicken (about 30 minutes). Spread mixture over crust in baking pan. Bake for 30 minutes, or until crust is browned and filling has thickened.

Table 14: Example generations under five entity-enhanced prompt types for a sample recipe from ReciFine. The left column lists entities and entity-enhanced prompt types; the right column shows extracted entities, ingredinet list or generated text.

Annotation Guideline for Recipe Evaluation

You will be asked to rate recipes on a 5-point Likert scale (1 = very poor, 5 = excellent) across several dimensions. Please read the recipe carefully and provide your judgment based on the descriptions below.

Scoring Scale (applies to all categories)

- 1 = Very Poor (fails completely, unacceptable)
- 2 = Poor (serious issues, hard to recommend)
- 3 = Fair (some issues, but still somewhat acceptable)
- 4 = Good (minor issues, overall strong)
- 5 = Excellent (meets the criteria perfectly)

Dimensions to Rate

1. Readability

- Does the recipe read fluently and naturally in English?
- Is it clear and easy to follow?
- Avoids confusing grammar, awkward phrasing, or missing steps.

2. Accuracy

- Are the listed ingredients, tools, chef actions correct?
- No major contradictions.
- The quantities and steps should make sense.

3. Feasibility

- Could this recipe realistically be prepared in a home kitchen?
- Are the steps possible to execute with normal cooking tools and methods?
- Avoids overly vague or impossible instructions.

4. Creativity

- Does the recipe show originality or novelty?
- Is it more interesting than a very standard or overly simple recipe?
- Does it add a unique twist, combination, or method?

5. Overall Quality

- Your overall impression of the recipe.
- Consider readability, accuracy, feasibility, and creativity together.
- Would you recommend this recipe to others?

Figure 9: Guide for annotating recipes, as provided in the Docanno annotation interface