Substance over Style: Document-Level Targeted Content Transfer

Allison Hegel[1]* **Sudha Rao**[2] **Asli Celikyilmaz**[2] **Bill Dolan**[2]
[1]Lexion, Seattle, WA, USA [2]Microsoft Research, Redmond, WA, USA
allison@lexion.ai {sudhra,aslicel,billdol}@microsoft.com

Abstract

Existing language models excel at writing from scratch, but many real-world scenarios require rewriting an existing document to fit a set of constraints. Although sentence-level rewriting has been fairly well-studied, little work has addressed the challenge of rewriting an entire document coherently. In this work, we introduce the task of document-level targeted content transfer and address it in the recipe domain, with a recipe as the document and a dietary restriction (such as *vegan* or *dairy-free*) as the targeted constraint. We propose a novel model for this task based on the generative pre-trained language model (GPT-2) and train on a large number of roughly-aligned recipe pairs.[1] Both automatic and human evaluations show that our model out-performs existing methods by generating coherent and diverse rewrites that obey the constraint while remaining close to the original document. Finally, we analyze our model's rewrites to assess progress toward the goal of making language generation more attuned to constraints that are substantive rather than stylistic.

1 Introduction

We often think that writing starts from a blank page, but in practice, writing often involves adapting an existing document to fit a new context. This might involve rewriting documentation written for a Mac so that it will apply to a PC, rewriting a lesson plan for a different grade level, or rewriting a product description to appeal to customers in multiple regions. Automating such rewriting is valuable but challenging, since it requires learning to make coordinated changes spanning an entire document while adhering to constraints that apply not to the style but to the substance of the document.

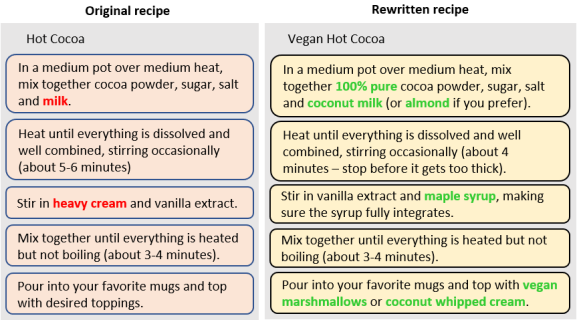

Figure 1: Document-level targeted content transfer in the recipe domain: given a hot cocoa recipe and the user constraint *vegan*, the task is to rewrite the recipe into a vegan hot cocoa recipe.

We introduce the novel task of **document-level targeted content transfer**, defined as rewriting a document to obey a user-provided constraint resulting in some systematic alteration of the document's content. Success at this task involves both transfer and controlled generation at the document level. Prior work on controlled generation guides the output of a model using attribute classifiers (Dathathri et al., 2020) or control codes (Keskar et al., 2019), but we find that these models do not perform well on our transfer task (§4.1.2). In contrast, models built for the transfer task are generally trained at the sentence level (Hu et al., 2017b,a; Li et al., 2018; Rao and Tetreault, 2018; Syed et al., 2019).

Document-level transfer has typically found success by rewriting each sentence independently (Maruf et al., 2019). However, many real-world rewriting scenarios require interdependent changes across multiple sentences. A clear example is cooking, where rewriting a hot cocoa recipe to make it *vegan* requires more than just substituting "coconut milk" for "milk" in a single step—it may also require changing the cooking times and techniques, adjusting ingredient amounts, or replacing other ingredients like toppings or spices (Figure 1). Such

*Work done when the author was at Microsoft Research.
[1]https://github.com/microsoft/document-level-targeted-content-transfer

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6485–6504,
November 16–20, 2020.

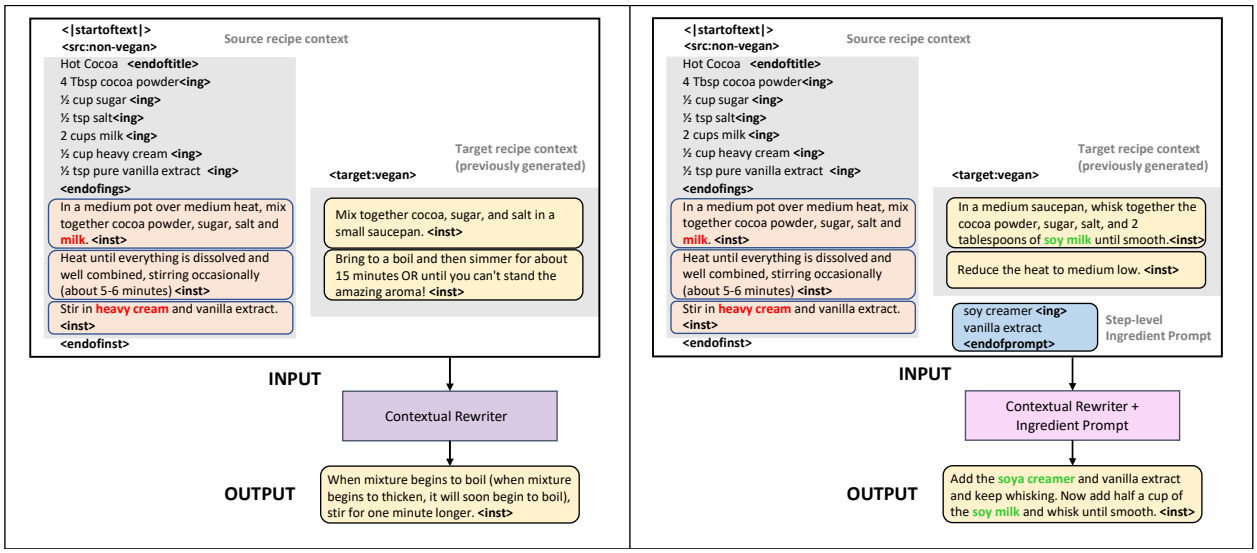

Figure 2: Rewrites of the source n^{th} step obtained by the two variants of our proposed model (at test time): (left) Contextual Rewriter, which uses the source context until the n^{th} step and the target context until the $(n-1)^{th}$ step to generate the target n^{th} step; and (right) Contextual Rewriter + Ingredient Prompt, which uses the same context as the previous variant with the addition of a step-level ingredient prompt.

a rewriting task is substantive rather than stylistic because it changes the content of the recipe, while a stylistic transfer on recipes might instead focus on rewriting a recipe for a different audience, reading level, or writing style such that the content remains the same and only the expression of the recipe changes.

In this work, we address the task of document-level targeted content transfer in the recipe domain, where the document is a recipe and the target constraint is a dietary restriction such as *vegan*. Given a recipe (source) and a dietary constraint, the task is to rewrite it into a new recipe (target) that obeys the constraint. Training a fully-supervised model for this task requires a large number of (recipe, rewritten recipe) pairs, which are difficult to obtain at scale. We therefore leverage an alignment algorithm (Lin et al., 2020) to construct our *noisy* training data pairs where the source is a recipe that violates a dietary constraint and the target is another recipe for the same dish that obeys the constraint but may not be similar to the source (§2).

We propose a novel model for this task which learns to rewrite a source document one step at a time using document-level context. We start with the recently successful generative pre-trained (GPT-2) language model (Radford et al., 2019) and fine-tune it on text that combines {document-level context, source step, constraint, target step} using appropriate separators. We investigate two variants of our model in the recipe domain:

Contextual Rewriter (§3.1) where the context includes the source recipe (including title, list of ingredients, and steps), any previously rewritten steps, and the targeted constraint (Figure 2 left);

Contextual Rewriter + Ingredient Prompt (§3.2) where, in addition to the context discussed above, we predict a set of step-level ingredients to prompt our rewriter model (Figure 2 right).

We compare our proposed models to sentence-level transfer baselines that rewrite each recipe step independently, and to document-level controllable baselines that ignore the source recipe and only control for the dietary constraint (§4.1). We use automatic metrics and human judgments to evaluate the rewritten recipes, measuring their overall quality, their fluency, their dietary constraint accuracy, and their ability to produce diverse outputs without straying too far from the source recipe (§4.2). Comprehensive experiments demonstrate that our proposed model outperforms baselines by simultaneously accomplishing both transfer and control, but still lacks the substantive knowledge humans rely on to perform well at this task (§4.5). Finally, we conduct an in-depth analysis of various model rewrites and the strengths and weaknesses of the models (§5).

2 Dataset Creation

The recipe domain, constrained by dietary restrictions, is particularly well-suited to our task since recipes are commonly rewritten according to dietary constraints in real-world scenarios[2], and this process often requires multiple related changes across the recipe. To construct our dataset, we use three steps: collect recipes spanning a range of dietary constraints (§2.1), tag recipes with dietary constraints using a rule-based method (§2.2), and align recipes into pairs with similar content but opposite dietary tags (§2.3).

Although our model relies on large amounts of parallel data, we obtain this parallel data automatically by running an unsupervised alignment algorithm (Lin et al., 2020) on non-parallel data. Large collections of non-parallel data are readily available on the web for many other domains, such as lesson plans for different grade levels or technical documentation for different operating systems. With the methods outlined in this section, non-parallel data can be aligned and transformed into a parallel dataset for transfer tasks in other domains.

2.1 Collect Recipes

We collect English recipes from online recipe websites.[3] We remove recipes that lack a title or a list of ingredients, or that have less than two steps. The resulting dataset contains 1,254,931 recipes, with a median of 9 ingredients and 9 steps.

2.2 Tag Recipes with Dietary Constraints

We consider seven dietary constraints: dairy-free, nut-free, egg-free, vegan, vegetarian, alcohol-free, and fish-free.[4] For each dietary constraint, we obtain a list of ingredients that violate it using food lists from Wikipedia.[5] We then compare each recipe's ingredients against that list, and tag it *valid* if there are no violating ingredients, or *invalid* if a violating ingredient is in the recipe.

Dietary Constraint	Recipe Pairs Train	Dev	Test	Step Pairs Train
Diary-Free	194,309	10,607	9,190	2,552,492
Nut-Free	161,596	8,722	8,989	2,060,228
Egg-Free	124,207	5,786	5,662	1,794,047
Vegan	110,718	5,708	4,859	1,765,865
Vegetarian	59,847	2,765	2,629	682,845
Alcohol-Free	52,157	2,348	2,136	570,627
Fish-Free	34,786	1,546	1,278	383,162

Table 1: Number of recipe pairs and step pairs for each dietary restriction in our data.

2.3 Create Recipe and Step Pairs

Our goal is to find recipe pairs for the same dish where one obeys a dietary constraint and the other violates it. Lin et al. (2020) propose a method for automatically aligning two recipes of the same dish. We use their method to first group recipes into dishes, and then find aligned pairs of recipes within a dish where one is valid and the other is invalid. Table 1 shows the number of recipe pairs in our dataset for each dietary constraint. It should be noted that these pairs are *noisy* for our rewrite task since the pairs were not created by rewriting.

The alignment algorithm also gives an alignment score at the step level. We threshold on this score to keep only the highest-quality step pairs. Further, in cases where a single source step is aligned to more than one target step with a high score, we combine the target steps together into one, enabling our rewrite model to learn to rewrite one step into multiple steps whenever appropriate. Table 1 (rightmost column) shows the total number of high quality step-level pairs for each dietary constraint that we use to train our rewrite model.

3 Model Description

We propose two model variants for document-level targeted content transfer in the recipe domain. Given a recipe and a dietary constraint, the goal is to rewrite the recipe one step at a time to fit the dietary constraint.

3.1 Contextual Rewriter

We start with a pre-trained GPT-2 model which is trained on text from 45 million websites with a language modeling objective to predict the next word given previous words.[6] We fine-tune this model

[2] In a survey of 250 randomly selected user comments from recipe websites, we found that one third discussed modifying the recipe, often to accommodate dietary restrictions. In addition, U.S. public school cafeterias are required by law to accommodate food allergies and other dietary needs (USDA, 2017). Such rewriting that is currently done manually could benefit from our proposed automated approach.

[3] Websites include Food.com, AllRecipes.com, FoodNetwork.com, and 8 other websites, as well as four existing recipe datasets. Appendix contains full list and associated statistics.

[4] Each of these constraints is commonly mentioned in recipe titles, and is one of the most common diets (USDA, 2020) or dietary restrictions (FDA, 2020).

[5] E.g. for the dairy-free constraint, we used `https://en.wikipedia.org/wiki/Dairy_product`.

[6] This and any future discussion of a pre-trained GPT-2 model refers to the GPT-2 medium model available at `https:`

using the same language modeling objective on the train split of step-level recipe pairs (Table 1). The left column of Table 2 shows how we format our pairwise data for fine-tuning. Given an aligned pair of a source step (n) and a target step (n'), we prepend the source step n with the source recipe's title, ingredients, and steps from 1 to $(n-1)$; we also prepend the target step n' with target steps from 1 to $(n'-1)$. We use separators to demarcate each piece of contextual information. Further, to allow the GPT-2 model to understand the dietary constraint, we prepend the entire source-level context with a special tag <src:non-constraint> (e.g. non-vegan) and prepend the entire target-level context with a special tag <tgt:constraint> (e.g. vegan).

Note that during fine-tuning we use only those steps of a recipe that have been aligned into a pair with a high alignment score (§2.3). However, at test time, we rewrite all steps in the source recipe using the fine-tuned model. Also, during fine-tuning, we use the teacher forcing strategy: while rewriting source step n, the target recipe context corresponds to the true target steps 1 to $(n'-1)$, whereas during test time, the target recipe context corresponds the previously generated steps 1 to $(n-1)$.[7]

3.2 Contextual Rewriter + Ingredient Prompt

We observe that the rewriter described above often uses ingredients and techniques that diverge from the source recipe. For example, on the left side of Figure 2, the rewritten output diverges from the source recipe when it ignores the ingredients of *"heavy cream and vanilla extract"* in the source step rather than suggesting an appropriate vegan alternative. We hypothesize that if the model had the capacity to accept step-level ingredients (in the form of a prompt) as an additional input while rewriting each step, then it could learn to follow the source recipe more closely. This strategy has proven effective in other domains, including automatic storytelling, where prompting a model with a rough "storyline" helps models stay on-topic (Yao et al., 2018).

We therefore propose a variant of the previous model that uses step-level ingredients as a prompt in addition to document-level context. We again start with a pre-trained GPT-2 model and fine-tune it on the train split of step-level recipe pairs (Table 1) using a different data format (see the right column of Table 2). As in the previous model, we use the source recipe data until step n and the target recipe steps until $(n'-1)$. But before including the target step n', we prompt with the ingredients in n' separated by an <ing> separator, and end with an <endofprompt> special token. This enables our model to learn to use the ingredient prompt while generating the rewrite.

We investigate two methods for generating the step-level ingredient prompt. During fine-tuning, we use the rule-based method. At test time, we generate results using both methods.

Rule-based ingredient prompt: Given a source recipe step, we first identify all ingredients mentioned in the step.[8] We then use a rule-based method to substitute any ingredients that violate the dietary constraint with alternatives from a food substitution guide (Steen and Newman, 2010). While there is work on automatically substituting recipe ingredients with similar ones (Teng et al., 2012; Boscarino et al., 2014; Yamanishi et al., 2015), to our knowledge no work makes recipe substitutions in accordance with dietary constraints.

< |startoftext| >	< |startoftext| >
<src:non-constraint>	<src:non-constraint>
src_title <endoftitle>	src_title <endoftitle>
src_ingredient 1 <ing>	src_ingredient 1 <ing>
...	...
src_ingredient K	src_ingredient K
<endofings>	<endofings>
src_step 1 <inst>	src_step 1 <inst>
...	...
src_step n	src_step n
<endofinst>	<endofinst>
<tgt:constraint>	<tgt:constraint>
tgt_step 1 <inst>	tgt_step 1 <inst>
...	...
tgt_step n'	tgt_step $(n'-1)$
<endofinst>	<endofinst>
< |endoftext| >	tgt_step n' ingredient 1 <ing>
	...
	tgt_step n' ingredient K'_n
	<endofprompt>
	tgt_step n'
	< |endoftext| >

Table 2: Data format for fine-tuning a GPT-2 model to rewrite source recipe step n into target recipe step n' (where n' is aligned to n) using our Contextual Rewriter (left) and our Contextual Rewriter + Ingredient Prompt (right).

//github.com/huggingface/transformers.

[7]For decoding, we use top-k sampling ($k = 40$). Appendix contains implementation details for all models.

[8]For each ingredient in the recipe's ingredient list, we find the longest n-gram match between ingredient and step, ignoring common recipe stopwords such as "tablespoons" and descriptors like "chopped."

GPT-2 ingredient prompt: We use a GPT-2 model to predict the step-level ingredients to use as prompts. We first collect a dataset of recipe steps from ∼1.2 million recipes (from §2.1). We extract the ingredients from each recipe step using the rule-based method above. We then construct texts by combining {recipe title, full list of ingredients, steps 1 to $n-1$, ingredients in step n} and fine-tune another GPT-2 model on this text.[9]

4 Experimental Results

We aim to answer the following research questions:

1. Do generation-based rewriters outperform simpler non-learning baselines (§4.1.1)?
2. Do our proposed rewriters do a better job of staying close to the source recipe while obeying the constraint compared to controllable generation models (§4.1.2) that obey the constraint but ignore the source recipe?
3. Do our proposed document-level rewriters outperform sentence-level rewriters (§4.1.3)?
4. Does using ingredients as a prompt help our proposed rewriter stay close to the source recipe while obeying the dietary constraint?
5. Finally, how do models compare to human performance on the rewrite task (§4.5)?

4.1 Baselines and Model Ablations

4.1.1 Non-learning Baselines

Rule-Based: We use the rule-based method discussed in §3.2 to rewrite each step independently. This baseline only substitutes ingredients and does not change the cooking times or techniques that may be required for the substitutions to fit.

Retrieval: We imitate a simple approach to the recipe rewrite task: searching the web for a version of the dish that obeys the given dietary constraint. Given a source recipe, we determine the dish to which this recipe belongs and retrieve a recipe for the same dish that fits the dietary constraint from the combined pool of train, dev, and test recipes.

4.1.2 Document-level Controllable Baselines

We build the following baseline models by providing the title and ingredient list of the target recipe (which obeys the dietary constraint) as the prompt to generate the first target recipe step. For generating each of the subsequent n^{th} steps, we append the previously generated steps 1 to $(n-1)$ to the prompt. We stop when the model has generated as many steps as there are in the source recipe.

PPLM: Plug-and-Play Language Model (Dathathri et al., 2020) combines a pre-trained language model with a classifier to guide the generation toward a user-specified attribute. We build a PPLM model for our task using a GPT-2 model fine-tuned on ∼1.2 million recipes (§2.1) as the pre-trained language model and using separate bag-of-words classifiers for each of our dietary constraints.[10]

CTRL: The conditional transformer language model (Keskar et al., 2019) uses a 'control' code to govern the style and content of the generated text. For our task, we use the "Links" control code to specify the recipe domain.[11]

4.1.3 Sentence-level Transfer Baselines

We build additional baseline models for rewriting each step independent of context and train them on our recipe step pairs (Table 1).

Seq2Seq Copy: We use a sequence-to-sequence model that is enriched with a copy mechanism (Jhamtani et al., 2017). We train separate models for each of our dietary constraints.

Transformer We train a transformer (Vaswani et al., 2017) model with byte-pair encoding.[12]

4.1.4 Model Ablations

No-Source Rewriter: We fine-tune a pre-trained GPT-2 model on ∼1.2 million recipes (from §2.1) with a simple language modeling objective. This ablation does not make use of the source recipe, but rather uses only the title and the ingredient list of the aligned target recipe as the prompt, generating the target recipe sequentially.

End-to-End Rewriter: This model variant is trained end-to-end to rewrite the entire source recipe at once rather than one step at a time. As a prompt, it takes a dietary constraint, a source recipe (title, ingredients and steps), and the title and ingredients of the target recipe. We start with a GPT-2 pre-trained model and fine-tune it on the train split of our recipe pair data (Table 1) for our task.

[9] Data format used for fine-tuning is included in appendix.

[10] See appendix for PPLM implementation details.

[11] See appendix for CTRL implementation details.

[12] We use the implementation at https://github.com/gooppe/transformer-summarization.

Model	Fluency Perplexity ↓	Dietary Constraint % Adherence ↑	Closeness to Source ROUGE ↑	Diversity Trigram ↑
Non-learning Baselines				
Rule-Based	10.24	96.1	*98.76*	0.550
Retrieval	**9.01**	93.4	28.40	0.344
Document-level Controllable Baselines				
PPLM	9.28	94.9	20.48	0.577
CTRL	13.47	94.3	24.69	0.418
Sentence-level Transfer Baselines				
Seq2seq Copy	15.60	99.0	25.98	0.145
Transformer	9.88	93.5	30.67	0.360
Model Ablations				
No-Source Rewriter	N/A	96.4	20.35	0.548
End-to-End Rewriter	9.51	97.0	25.60	0.488
No-Context Rewriter	10.79	**99.9**	31.81	0.615
Contextual Rewriter	11.61	**99.6**	31.16	0.634
+ GPT-2 Ingredient Prompt	13.86	**99.6**	28.93	0.590
+ Rule Ingredient Prompt	12.54	**99.5**	**34.06**	**0.674**

Table 3: Automatic metric results on model rewrites of 1000 randomly sampled recipes from the test set. The difference between bold and non-bold numbers is statistically significant with $p < 0.001$. We do not compare to *Rule-Based* under closeness to source since it copies steps from the source, leading to an artificially high score.

No-Context Rewriter: This variant does not make use of the document-level context, but rather learns to rewrite using only (source step, target step) pairs.

Contextual Rewriter: This variant makes use of document-level context, but does not use a step-level ingredient prompt.

Contextual Rewriter + GPT-2 Prompt: At test time, in addition to document-level context, this variant uses the GPT-2 step-level ingredient prediction model (§3.2) to generate an ingredient prompt.

Contextual Rewriter + Rule Prompt: This variant uses the rule-based method (§3.2) to generate an ingredient prompt.

4.2 Evaluation Metrics

4.2.1 Automatic Metrics

We evaluate model rewrites on 1000 recipes each from the test and dev sets on these criteria:

Fluency: We measure the perplexity of the model-generated recipes using a GPT-2 language model fine-tuned on recipe data for fair comparison.[13]

Dietary constraint accuracy: We report the percentage of ingredients in the rewritten recipes that obey the dietary constraint.[14]

Closeness to source:[15] We report ROUGE-L (Lin and Hovy, 2002) recall score between the source recipe and the rewritten recipe.

Diversity: Since generation models can produce results that are bland and repetitive, we measure the diversity of the generated recipes in terms of the proportion of unique trigrams (Li et al., 2015).

4.2.2 Human Judgments

We conduct human-based evaluation using a crowdsourcing platform[16] on rewrites from the best-performing models based on automatic metrics. We randomly sample 150 recipes from our test set with equal proportions of each dietary constraint.

Individual: We ask 5 judges to rate each rewritten recipe on a scale of 1 to 5 on these criteria:

a. Ingredient usage: *"Does this recipe use appropriate ingredients for the type of dish it is making?"*

b. Closeness to source: *"How close is this recipe to the source while fitting the dietary constraint?"* While some difference from the source is necessary for the rewriting task, this metric evaluates whether the recipe has strayed so far from the source that it may no longer be considered a rewriting of the source recipe.

[13] We do not report perplexity for the No-Source Rewriter since we use that model to calculate perplexity.

[14] To identify all ingredients in a recipe, we match against a list of foods from https://foodb.ca/.

[15] Note that we do not measure closeness to target since we do not have gold target rewritten recipes.

[16] We use https://www.mturk.com/. Details on selection, questions, design, and payment in appendix.

c. Dietary constraint: *"Does this recipe fit the specified dietary constraint?"*

d. Overall quality: *"Is this a good recipe for someone who follows this dietary constraint?"* We expect this metric to indirectly reflect qualities for which there are no well-accepted automatic metrics, such as coherence and the appropriateness of the ingredient prompts.

Comparative: We also collect human judgments on head-to-head comparisons between models by displaying two rewrites of the same source recipe side by side: one from our best-performing model (Contextual Rewriter + Rule Prompt) and the other from one of the Rule-Based, Retrieval, End-to-End Rewriter, or Contextual Rewriter models. We ask them to choose which of the two rewrites is better overall. Each pairwise comparison is rated by five judges.

4.3 Automatic Metric Results

While each model has its strengths, our proposed models provide the best balance of both transfer and control. Table 3 shows the results on model rewrites of 1000 randomly sampled recipes from the test set.[17] The retrieval baseline produces the most fluent rewrites, which is expected given that its outputs consist of human-written recipes. However, its scores for closeness to source and adherence to the dietary constraint are considerably lower. Document-level controllable baselines produce more diverse outputs than sentence-level transfer baselines, but sentence-level transfer baselines stay closer to the source recipe. In particular, Seq2seq Copy achieves a high dietary constraint accuracy, but we noticed that this model generates bland and repetitive outputs (as reflected in its diversity score). Each of these models has a shortcoming in a key component of the rewrite task.

Under our model ablations, we find that the No-Source Rewriter earns the lowest score for closeness to source, which is predictable given that it does not see the source recipe. By introducing source context, the End-to-End Rewriter does slightly better, producing fluent rewrites but still lacking diversity and dietary constraint accuracy. By rewriting each step independent of context, the No-Context Rewriter achieves a very high dietary constraint accuracy, but does not stay as close to the source as variants that use context. The model that introduces a GPT-2 predicted ingredient prompt obeys the dietary constraint well, but is not able to maintain diversity while staying close to the source, suggesting that there is room for improvement in how we build our ingredient prediction model. Finally, the rewriter that uses context and a rule-based ingredient prompt performs best across dietary constraint accuracy, closeness to source, and diversity while remaining reasonably fluent.

[17] Results on 1000 recipes from the dev set are reported in the appendix. They follow the same pattern as the test set.

Model	Ingredient Usage	Dietary Const.	Close to Source	Overall Quality
Rule-Based	4.64	4.70	*4.58*	4.47
Retrieval	4.48	4.40	3.29	3.91
End-to-End	4.64	4.72	3.73	4.52
Contextual	**4.71**	**4.74**	3.84	**4.60**
+ Rule Prompt	4.67	**4.75**	**4.06**	4.57

Table 4: Human judgments on a scale of 1 to 5 on model rewrites of 150 recipes from test set.

4.4 Human Judgment Results

Table 4 shows the results of human judgments on 150 recipe rewrites from the test set.[18] We find that all models except the retrieval baseline achieve similarly high scores. The Contextual Rewriter + Rule Prompt, the best-performing variant of our model according to automatic metrics, performs well in closeness to source and diversity, reaffirming our previous findings.[19] Interestingly, the Contextual Rewriter without an ingredient prompt performs better at ingredient usage and receives the highest overall score. Upon further investigation, we find that the rule-based method we used to generate the ingredient prompt sometimes suggests awkward ingredient substitutions such as *"goat soymilk"*, which leads to a lower ingredient usage score.

Figure 3 shows the results of model comparisons.[20] We find that humans prefer our best model considerably over the retrieval baseline, but the Rule-Based method and the End-to-End Rewriter come close to our best model. The Contextual Rewriter performs similarly to our best model.

4.5 Comparison to Human Rewrite

We ask three experienced cooks who are current or former vegetarians to rewrite 30 randomly sampled non-vegetarian recipes from our test set into vegetarian recipes. We find that the human rewrites

[18] Inter-annotator agreement (Krippendorff's alpha) is 0.12.

[19] As with automatic metrics, we do not compare to Rule-Based in closeness to source since it copies from the source.

[20] Inter-annotator agreement (Krippendorff's alpha) is 0.14.

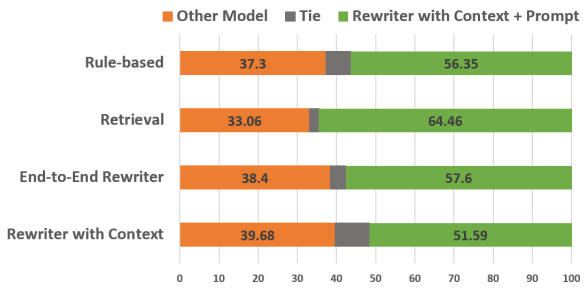

Figure 3: Results of a pairwise comparison between rewrites of our best model and other models on 150 recipes from the test set as judged by human evaluators.

significantly exceed our best model's performance in all four automatic metrics: fluency (perplexity: 13.91 vs. 20.8), adherence to the dietary constraint (99.7% vs. 96.3%), closeness to the source (ROUGE: 77.08 vs. 35.44), and diversity (0.908 vs. 0.836). These findings suggest that there is room for further improvement on this task.

5 Analysis

Simple substitution is not adequate for the task of document-level targeted content transfer. In a recipe that contains a single violating ingredient *"meat"*, the rule-based method makes the minimal edit of substituting *"imitation meat"*, but ignores the other parts of the recipe that must change as a result. Although on automatic metrics our model does only marginally better, qualitatively we found many cases where the rule-based method fails: it always suggests the same substitutions independent of the type of recipe leading to awkward food combinations, it misses a long tail of uncommon ingredients, and it does not make contextual changes to ingredient amounts, cooking times, or techniques. These flaws lead to the rule-based method performing worse than our model according to human judges (Table 4 and Figure 3).

As Figure 4 shows, the Contextual Rewriter + Rule Prompt is capable of more extensive changes based on document-level context. Human evaluators preferred our model's output, which changes multiple ingredients, adds additional techniques, and increases the cooking time. In general, while many of the baseline models tend to produce generic outputs such as *"Preheat the oven"*, our model produces much more diverse recipes and ingredient substitutions.

The larger the number of invalid ingredients for a dietary constraint, the more difficult it was for our model to follow that constraint. Vegan, the most restrictive constraint we studied, had the lowest dietary adherence accuracy across all models (93.6%). The alcohol-free constraint, which is dominated by one common ingredient (wine), had the highest accuracy (99.5%) despite the models seeing fewer training examples for that constraint.[21]

The Contextual Rewriter + Rule Prompt falls short in its understanding of the physical entities involved in cooking. Some of the steps it generates are not physically possible, such as *"Dip the cheese into the bread"*. The model can also suggest unrealistic or illogical cooking times (e.g. *"Bake for 10-10 minutes"*), or change oven temperature mid-recipe. While these results are uncommon, they highlight that the model has not learned the physical rules governing the use of ingredients and cooking techniques.

6 Related Work

Text attribute transfer: Most work in text attribute transfer has been at the sentence level, including sentiment (Hu et al., 2017b), formality (Rao and Tetreault, 2018), tense (Hu et al., 2017a), and authorship (Syed et al., 2019). While the text attribute transfer approach works well at the sentence level, our work tests a model's ability to make changes across multiple sentences that result in a coherent document. Further, our method allows for more drastic alterations to the source text than edit-based methods (Li et al., 2018) since we do not restrict the words that the model can alter.

Document-level controlled generation: The difficulty of text attribute transfer is amplified when the task expands to the document level. While controlled generation models such as Grover (Zellers et al., 2019), PPLM (Dathathri et al., 2020), and CTRL (Keskar et al., 2019) have been successful at the document level, they do not make use of context beyond an initial prompt. For many NLP tasks, contextual information from surrounding sentences can improve the quality of a generated sentence. We have seen this for coreference resolution (Joshi et al., 2019), relation extraction (Tang et al., 2020), and machine translation (Werlen et al., 2018; Macé and Servan, 2019). In this work, we show the effectiveness of including document-level context when rewriting recipes to fit a dietary constraint.

[21] Appendix shows a breakdown of each model's accuracy by dietary restriction.

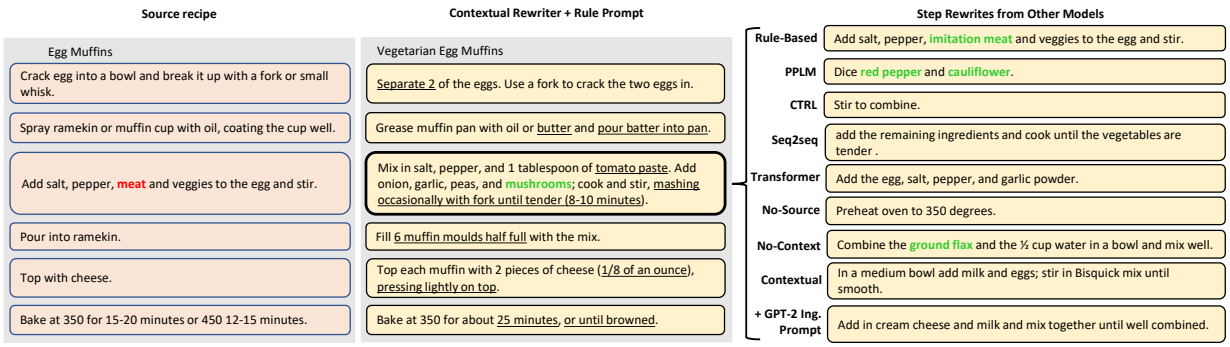

Figure 4: A recipe rewritten by the Contextual Rewriter + Rule Prompt, with outputs for a single step from other models for comparison. Our model replaces the violating ingredient (in **red**) with a substitution (in **green**), as well as modifying or adding new ingredients and techniques in every step (underlined).

Recipe generation: Recipe generation has been a research focus for decades, using methods ranging from rule-based planning systems (Hammond, 1986) to more recent neural network models that use targeted information such as entity types (Parvez et al., 2018), cooking actions (Bosselut et al., 2017), ingredients (Kiddon et al., 2016), or order information (Bosselut et al., 2018) to guide the generations. Building on the insight that knowledge about ingredients improves recipe generation, our work uses ingredient prompts to guide the generation of each recipe step. While there has been extensive work on recipe generation, few studies focus on controlled recipe generation. Majumder et al. (2019) recently introduced the task of personalized recipe generation, producing customized recipes based on user preferences. To our knowledge, our work is the first to generate recipes that conform to a given dietary constraint.

7 Conclusion

We introduce the novel task of document-level targeted content transfer and address it in the recipe domain, where our documents are recipes and our targeted constraints are dietary restrictions. We propose a novel model for rewriting a source recipe one step at time by making use of document-level context. Further, we find that conditioning the model with step-level constraints allows the rewritten recipes to stay closer to the source recipe while successfully obeying the dietary restriction. We show that our proposed rewriter is able to outperform several existing techniques, as judged both by automatic metrics and human evaluators.

Although we focus on the recipe domain, our method naturally generalizes to other domains where procedural tasks can be substantively rewritten. For example, one could rewrite technical documentation by constraining on the target operating system, rewrite lesson plans by constraining on the target grade level, or rewrite furniture assembly instructions by constraining on the tools used.

More broadly, this approach makes it possible to customize existing content to better fit a user's physical reality, whether that entails accommodating their dietary needs, updating their schedule based on the weather forecast, or providing information on a dashboard based on what's in their field of view. As language generation becomes more grounded in signals outside of language, work in the area of substantive transfer becomes increasingly relevant.

Acknowledgments

This work would not have been possible without the support of Microsoft Research's AI Residency program, particularly Becky Tucker and the 2019-20 cohort of residents who maintained a tight-knit research community despite our physical distance. We would also like to thank Sam Leiboff, Zach Price, and Matt Runchey for their feedback on the recipe content transfer task that informed our modeling approach. Finally, we are grateful to our anonymous reviewers for their thoughtful comments and suggestions.

References

Corrado Boscarino, Vladimir Nedović, Nicole J. J. P. Koenderink, and Jan L. Top. 2014. Automatic extraction of ingredient's substitutes. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, page 559–564, New York, NY, USA. Association for Computing Machinery.

Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po sen Huang, and Yejin Choi. 2018. Discourse-aware neural rewards for coherent text generation. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2017. Simulating action dynamics with neural process networks. *CoRR*, abs/1711.05313.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Epicurious. Epicurious - recipes with rating and nutrition. https://www.kaggle.com/hugodarwood/epirecipes. Accessed: 2020-05-31.

FDA. 2020. What you need to know about food allergies.

Kristian J. Hammond. 1986. Chef: A model of case-based planning. In *AAAI*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017a. Controllable text generation. *CoRR*, abs/1703.00955.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017b. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 329–339, Austin, Texas. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer.

Angela S Lin, Sudha Rao, Asli Celikyilmaz, Elnaz Nouri, Chris Brockett, Debadeepta Dey, and Bill Dolan. 2020. A recipe for creating multimodal aligned datasets for sequential tasks. *arXiv preprint arXiv:2005.09606*.

Chin-Yew Lin and Eduard Hovy. 2002. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4*, pages 45–51. Association for Computational Linguistics.

Loginetics. Now You're Cooking! recipe software. http://www.ffts.com/recipes.htm. Accessed: 2020-05-31.

Valentin Macé and Christophe Servan. 2019. Using whole document context in neural machine translation.

Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating personalized recipes from historical user preferences. In *EMNLP*, pages 5975–5981.

Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2019. A survey on document-level machine translation: Methods and evaluation.

Dimitri Merejkowsky. 2020. PyEnchant: Python bindings for the Enchant spellchecking system.

Md Rizwan Parvez, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2018. Building language models for text with named entities. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2373–2383, Melbourne, Australia. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Sudha Rao and Joel R. Tetreault. 2018. Dear sir or madam, may I introduce the YAFC corpus: Corpus, benchmarks and metrics for formality style transfer. *CoRR*, abs/1803.06535.

C. Steen and J.M. Newman. 2010. *The Complete Guide to Vegan Food Substitutions: Veganize It! Foolproof Methods for Transforming Any Dish into a Delicious New Vegan Favorite*. Fair Winds Press.

Bakhtiyar Syed, Gaurav Verma, Balaji Vasan Srinivasan, Anandhavelu N, and Vasudeva Varma. 2019. Adapting language models for non-parallel author-stylized rewriting.

Hengzhu Tang, Yanan Cao, Zhenyu Zhang, Jiangxia Cao, Fang Fang, Shi Wang, and Pengfei Yin. 2020. Hin: Hierarchical inference network for document-level relation extraction. *arXiv preprint arXiv:2003.12754*.

Chun-Yuen Teng, Yu-Ru Lin, and Lada A. Adamic. 2012. Recipe recommendation using ingredient networks. In *Proceedings of the 4th Annual ACM Web Science Conference*, WebSci '12, page 298–307, New York, NY, USA. Association for Computing Machinery.

USDA. 2017. 2017 edition of accommodating children with disabilities in the school meal programs.

USDA. 2020. Dietary guidelines for americans: 2015-2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Lesly Miculicich Werlen, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. *CoRR*, abs/1809.01576.

Ryosuke Yamanishi, Naoki Shino, Yoko Nishihara, Jun ichi Fukumoto, and Aya Kaizaki. 2015. Alternative-ingredient recommendation based on co-occurrence relation on recipe database. In *KES*.

Lili Yao, Nanyun Peng, Ralph M. Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2018. Plan-and-write: Towards better automatic storytelling. *CoRR*, abs/1811.05701.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *NeurIPS*.

A Dataset Creation

We collect recipes from recipe websites and existing recipe datasets listed in Table 5.

While some websites use tags to indicate that a recipe obeys a dietary constraint, not all do, and the tags are often noisy or missing. We therefore choose not to rely on recipe websites for these tags, and instead we use a rule-based method to tag recipes in our dataset as either *valid* or *invalid* in relation to a dietary constraint. While the method improves our model's performance, we observe several shortcomings. Despite constructing a large set of rules, we still miss words that are uncommon or that did not appear in the train set. Also, since we search for invalid ingredients using the recipe's list of ingredients, we miss ingredients that have been omitted from the ingredient list, as well as ingredients that are not mentioned explicitly by name (e.g. *"fillet"* as in *"catfish fillet"* will not be flagged as an invalid ingredient for a fish-free recipe) or ingredients that are referred to by a brand name or slang term that is not part of our rule set.

Recipe Website	Number of Recipes
AllRecipes.com	58,535
BBCGoodFood.com	9,171
Chowhound.com	3,890
CommonCrawl	424,621
Epicurious.com (Epicurious)	20,110
Food52.com	20,595
Food.com	268,914
FoodNetwork.com	47,187
Instructables.com	11,190
MasterCook (Loginetics)	72,141
MealMaster (Loginetics)	312,344
ShowMeTheYummy.com	555
SimplyRecipes.com	2,372
SmittenKitchen.com	986
WikiHow.com	2,320

Table 5: Online recipe data sources and amounts.

While we tried to catch as many of these cases as possible, there are many ambiguous words that the method will incorrectly classify such as *"beefsteak tomato"* appearing to contain meat (*"steak"*), *"oyster crackers"* appearing to contain fish (*"oyster"*), or a variety of *"egg replacer"* brand-name products appearing to contain egg.

The method is also unable to recognize negation (e.g. *"This recipe is not vegan!"*), or distinguish when a food is marked as optional or as an alternative (e.g. *"Flax is a good substitute for eggs"*). Both of these situations would cause a recipe to be marked with the wrong tag.

After assigning tags, we align similar recipes to form pairs of recipes for the same dish. Table 6 shows an example alignment between two recipes for Hot Cocoa with the alignment scores for each step. Recipes were divided into 80% train, 10% dev, and 10% test sets before aligning them into pairs, resulting in slightly uneven sizes for each set.

B GPT-2 Model Details

For each GPT-2 model, we use the 355 million parameter pre-trained GPT-2 medium model. We fine-tune using batch sizes ranging from 2-16 distributed across 64 NVIDIA Tesla V100 GPUs. We use a block size of 1024 for the end-to-end rewriter, and smaller block sizes for models that generate one step at a time of 128 for models without context and 256 for models with context. We train

ID	Source Recipe Steps	ID	Target Recipe Steps	Score
0	In a medium pot over medium heat, mix together cocoa powder, sugar, salt and milk.	0	Heat milk to your desired temperature.	10.0
		1	While milk is being heated, mix hot cocoa mix, creamer, and cinnamon sugar in bowl.	99.7
1	Heat until everything is dissolved and well combined, stirring occasionally (about 5-6 minutes).	2	Add small squirt or about 1/4 teaspoon of chocolate syrup to dry mix.	1.0
2	Stir in heavy cream and vanilla extract.	3	Add same amount of syrup again, or enough so that dry mix becomes lumps.	37.0
3	Mix together until everything is heated but not boiling (about 3-4 minutes).	4	Add confectioner's sugar and cocoa powder to mix (doesn't have to be as lumpy anymore).	1.1
4	Pour into your favorite mugs and top with desired toppings.	5	Pour mix into mug and pour milk on top.	99.9
		6	Add whipped cream and extra chocolate syrup.	87.4

Table 6: Step-level alignment scores between two Hot Cocoa recipes from the dataset.

each model for 2 epochs on datasets of aligned recipe steps ranging from 1.4 million to 10 million instances. The No-Context Rewriter was the fastest model to train, at 26 hours per epoch, and the slowest were the End-to-End Rewriter and the Contextual Rewriter + Rule Prompt at 318 hours per epoch.

We experimented with several hyperparameters for generation, including top-k sampling, nucleus sampling, and temperature (Table 7) using manually-chosen values. Since most variants performed well in adherence to the dietary constraint, we chose the best-performing variant in perplexity and diversity for our experiments.

We observe that our models can generate diverse rewrites from the same prompt, each with a different degree of fluency and adherence to the dietary constraint. We therefore create a set of rules to select the best generation out of 10 using a set of criteria including use of invalid ingredients, non-dictionary words, and incorrect punctuation. The criteria for selecting from multiple generations include:

- The step does not contain any violating ingredients
- The length is less than 100 characters
- The step does not contain special characters including '%', '*', or '$'.
- The first character is capitalized
- The last character is punctuation
- All words appear in an English dictionary (Merejkowsky, 2020)

C Data Format for Document-Level Controllable Baselines

PPLM We use the official codebase for PPLM: https://github.com/uber-research/PPLM. To build our PPLM model on our datasets, we use a pre-trained GPT-2 model on ∼1.2 million recipes as the pre-trained language model. We build separate bag-of-words classifiers for each of our seven dietary constraints. We construct the bag-of-words for each dietary constraint by selecting words that appear at least 5 times in recipes fitting the constraint and do not appear in recipes that violate the constraint. At test time, we format the data with the same separators for title, ingredients, and steps used to fine-tune the GPT-2 model on recipe data.

CTRL For our task, we use the "Links" control code to specify the recipe domain. We include the desired dietary restriction in the prompt in addition to the target recipe context and separate them by newlines as they would appear in a web link. We also append the appropriate step number (e.g. "1.") to the prompt before generating each step.

D Data Format for Model Ablations

We format our recipe data differently for each model ablation described in the main paper. Table 8 shows the data format we use to fine-tune the GPT-2 model that predicts the ingredients in the next step. Table 9 shows the data format we use to fine-tune the End-to-End Rewriter. Table 10 shows the data format we use to fine-tune the No-Context Rewriter. Finally, Table 11 shows the data format we use to fine-tune the Contextual Rewriter.

E Example Outputs

Figure 5 shows a source recipe alongside the recipe generated by the Contextual Rewriter + Rule Prompt, as well the generated fourth recipe step from each other model for comparison.

We provide additional step-level examples for each model in Table 12, and examples of an entire recipe rewrite for each model in Table 13. We also

	Perplexity	% Adherence	Trigram Diversity
Contextual Rewriter + Rule Prompt			
top k = 40, nucleus = 1, temperature = 1	**12.54**	99.5	**0.709**
top k = 40, nucleus = 0.8, temperature = 1	12.57	95.1	0.472
top k = 40, nucleus = 0.9, temperature = 1	12.82	94.8	0.498
top k = 40, nucleus = 1, temperature = 0.9	14.13	94.0	0.526
top k = 0, nucleus = 1, temperature = 1	17.02	99.3	0.551
top k = 10, nucleus = 1, temperature = 1	13.98	99.3	0.492
top k = 20, nucleus = 1, temperature = 1	14.85	**99.6**	0.511

Table 7: Results on the dev set for various generation hyperparameters, including top-k sampling, nucleus sampling, and temperature.

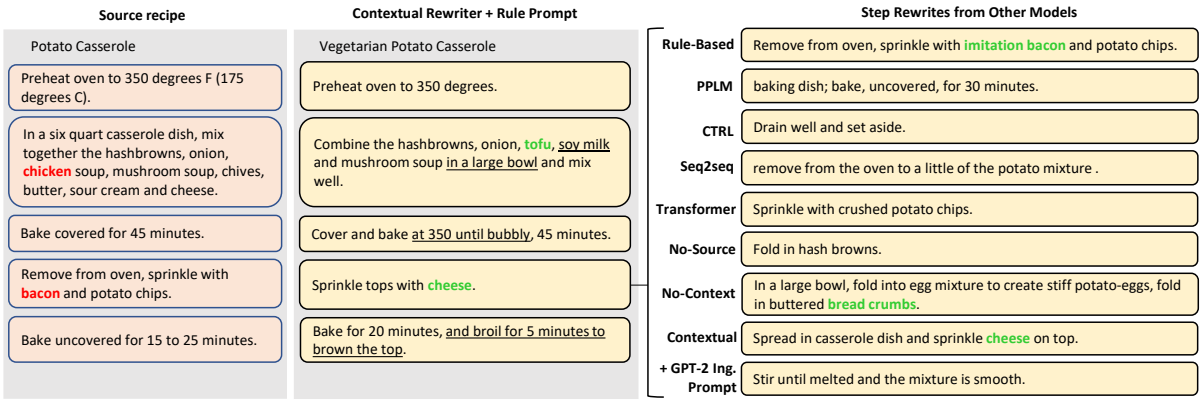

Figure 5: A recipe rewritten by the Contextual Rewriter + Rule Prompt, with outputs for a single step from other models for comparison. Our model replaces the violating ingredient (in **red**) with a substitution (in **green**), as well as modifying or adding new ingredients and techniques in every step (underlined).

< |startoftext| >
title <endoftitle>
ingredient 1 <ing>
...
ingredient K <ing>
<endofings>
step 1 <inst>
...
step $(n-1)$
<endofinst>
step n ingredient 1 <ing>
...
step n ingredient K_n
< |endoftext| >

Table 8: Data format used to fine-tune a GPT-2 model to predict the ingredients in the next step. If there were no ingredients in the next step, we used the token <noings>.

< |startoftext| >
<src:non-constraint>
src_title <endoftitle>
src_ingredient 1 <ing>
...
src_ingredient K <ing>
<endofings>
src_step 1 <inst>
...
src_step N
<endofinst>
<tgt:constraint>
tgt_title <endoftitle>
tgt_ingredient 1 <ing>
...
tgt_ingredient K <ing>
<endofings>
tgt_step 1 <inst>
...
tgt_step N
<endofinst>
< |endoftext| >

Table 9: Data format used to fine-tune the End-to-End Rewriter.

< |startoftext| >
<src:non-constraint>
src_title <endoftitle>
src_step N
<endofinst>
<tgt:constraint>
tgt_step N
< |endoftext| >

Table 10: Data format used to fine-tune the No-Context Rewriter.

< |startoftext| >
<src:non-constraint>
src_title <endoftitle>
src_ingredient 1 <ing>
...
src_ingredient K <ing>
<endofings>
src_step 1 <inst>
...
src_step $(n-1)$
<endofinst>
<tgt:constraint>
tgt_step N
< |endoftext| >

Table 11: Data format used to fine-tune the Contextual Rewriter.

show several examples of the ingredient prompts and resulting generations for our two prompt-based models, Contextual Rewriter + GPT-2 Prompt and Contextual Rewriter + Rule Prompt (Table 14).

F Additional Results

We provide the automatic metric results for 1000 recipes randomly sampled from the dev set in Table 15. We also provide a detailed breakdown of each model's accuracy across the seven dietary constraints in Table 16. Finally, we show a comparison of the results for human-written recipe rewrites against our best model, the Contextual Rewriter + Rule Prompt, on a subset of 30 vegetarian recipes from the test set (Table 17).

G Human Evaluation

For human evaluation, we limited our annotators to workers who met the following criteria:

- HIT Approval Rate (%) for all Requesters' HITs greater than 90
- Location is one of AU, CA, NZ, GB, US
- Number of HITs Approved greater than 500
- Masters has been granted (user was identified by the platform as a high-performing annotator)

We obtained 5 evaluations per recipe for each of the questions listed in Figure 6 (paying $0.30 per response), Figure 7 ($0.25), and Figure 8 ($0.50). For the head-to-head model comparison, if fewer than 3 of the 5 evaluations agreed, we considered it a tie between the models. We did not have our human annotators evaluate the fish-free dietary constraint since the most common violating ingredient, Worcestershire sauce, is not commonly known to contain fish, which caused our annotators confusion in an initial test run.

Model	Ultimate Green Beans (Vegetarian)
Original Recipe Step	Cook the bacon in a large, deep skillet over medium-high heat until crisp, about 10 minutes.
Rule-Based	Cook the imitation bacon in a large, deep skillet over medium-high heat until crisp, about 10 minutes.
PPLM	Saute mushrooms in butter until lightly browned.
CTRL	Place the mushrooms, butter, and sugar in a large pot.
Seq2seq Copy	heat oil in a large pot over medium heat .
Transformer	In a large skillet, heat the oil.
No-Source Rewriter	Saute mushrooms in butter until soft; set aside.
No-Context Rewriter	WEST FRONT STREET, ST. LOUIS.
Contextual Rewriter	Saute the diced onion in the olive oil over medium heat until golden around the edges.
Contextual Rewriter + GPT-2 Prompt	Chop a bunch of mushrooms, and slice a small onion.
Contextual Rewriter + Rule Prompt	In large skillet, over medium heat, saute mushrooms in butter 5 minutes.
Human Rewrite	Peel and sautee garlic and olive oil in large, deep skillet over high heat until slightly scorched skin, about 3 minutes.
	Vegetable Stock (Dairy-Free)
Original Recipe Step	Melt butter in 8-quart pan over medium heat.
Rule-Based	Melt nondairy butter in 8-quart pan over medium heat.
PPLM	Strain through a fine sieve.
CTRL	Add the carrots, celery stalks, green onions, garlic cloves and 1 cup of water.
Seq2seq Copy	heat oil in a large saucepan over medium heat .
Transformer	In a large saucepan, combine the stock, water, and seasonings.
No-Source Rewriter	Clean and half the vegetables
No-Context Rewriter	Heat the olive oil in a large pot over medium heat.
Contextual Rewriter	Heat the oil in a large 4-quart stockpot over medium heat.
Contextual Rewriter + GPT-2 Prompt	Step 4 Then add the trimmings from the carcass and 1 onion and a turnip, and the carrots and celery, and cook until the vegetables are soft, around 4 to 5 hours on a medium heat.
Contextual Rewriter + Rule Prompt	Feel free to play with this amounts; for a cheesy flavour, you can add vegan cheese or sprinkle nutritional yeast on top.
	Low Cholesterol Muffins (Egg-Free)
Original Recipe Step	In another bowl, beat the egg white and mix together with a fork rapidly for 30 seconds.
Rule-Based	In another bowl, beat the egg white substitute and mix together with a fork rapidly for 30 seconds.
PPLM	Bake in a preheated oven at 375F for 15
CTRL	Bake for 20-25 minutes till golden brown.
Seq2seq Copy	in a large bowl , combine the flour , sugar , baking powder , and salt .
Transformer	Bake at 350 degrees for 30 minutes.
No-Source Rewriter	Bring to boil water/juice.
No-Context Rewriter	Mix together dry ingredients.
Contextual Rewriter	In a small bowl, beat egg replacer with water until fluffy.
Contextual Rewriter + GPT-2 Prompt	Combine 1T globs of the mix onto greased cookie sheets.
Contextual Rewriter + Rule Prompt	Then add egg or egg replacer.
	N'Awlins Stuffed Bell Peppers (Fish-Free)
Original Recipe Step	Add shrimp, and cook for 2 minutes.
Rule-Based	Add tofu, and cook for 2 minutes.
PPLM	Makes 8 to 10 servings.
CTRL	To serve, place on serving platter and top each portion with another layer of vegetables.
Seq2seq Copy	add the shrimp and cook for 5 minutes .
Transformer	Add the rice and cook, stirring occasionally, until the rice is tender
No-Source Rewriter	Heat the oven to 350F.
No-Context Rewriter	Simmer covered until the peppers are tender.
Contextual Rewriter	Add in the minced garlic and red pepper flakes.
Contextual Rewriter + GPT-2 Prompt	Add rice and simmer just until the liquid has been absorbed.
Contextual Rewriter + Rule Prompt	Add the tofu and cook for 8 minutes.

Table 12: Example step outputs for different dietary constraints. Ingredients that violate the dietary constraint and their substitutions are colored.

Model	Caramel Popcorn (Dairy-Free)
Original Recipe	Melt butter, corn syrup, and brown sugar together. Add sweetened condensed milk slowly. Cook to soft ball stage. Pour over popcorn and mix.
Rule-Based	Melt nondairy butter, corn syrup, and brown sugar together. Add sweetened condensed soymilk slowly. Cook to soft ball stage. Pour over popcorn and mix.
Retrieval	In a large saucepan combine brown sugar, corn syrup, and baking soda. Stir to combine and bring to a soft boil. Add vinegar and boil again. Remove from heat and add vanilla. Place popcorn in a large baking pan. Pour sauce over popcorn and coat evenly. Cover with plastic wrap to keep fresh. Fill brown bags with a scoop of caramel popcorn for take home treats!
PPLM	Combine sugar, margarine, syrup and salt in a heavy pan. Stir over low heat until mixture is melted and boiling. Boil for 5 minutes, stirring occasionally. Remove from heat.
CTRL	Preheat oven to 350 degrees F. In a large bowl, combine the cornstarch, coconut oil, and salt. Add the peanuts, sugar, salt, vanilla, and baking soda. Stir until well combined.
Seq2seq Copy	in a large bowl , combine the sugar , brown sugar , and vanilla . add the egg and vanilla extract . cook on low for about 5 minutes . pour into a large bowl and set aside .
Transformer	In a large saucepan, combine brown sugar, corn syrup, and margarine In a large saucepan, combine the brown sugar, corn syrup, and Place popcorn in a large, buttered baking pan. Pour over popcorn and toss to coat.
End-to-End Rewriter	Pop popcorn. Remove any un-popped kernels. Put popcorn in a large bowl. Put the nuts on top. In a saucepan, melt margarine, brown sugar, corn syrup and salt. Bring to a boil, stirring constantly. Boil without stirring for 5 minutes. Remove from heat and add vanilla and soda. Pour syrup over popcorn and mix well. Bake in a preheated 200 degree oven for 1 hour, stirring every 15 minutes.
No-Context Rewriter	Combine margarine, Kahlua, brown sugar, corn syrup and salt in a large, nonstick saucepan. Heat margarine, brown sugar and corn syrup. Boil for 5 minutes over medium heat without stirring. Stir; bake 5 minutes more. Stir; bake 5 minutes longer. Pour caramel over popcorn and stir well. Spread evenly onto cookie sheet 4 Bake 45 to 55 minutes or until golden brown. Cool completely, about 15 minutes.
Contextual Rewriter	In a saucepan, mix brown sugar, margarine, corn syrup & salt. Add 2 cups maple syrup, salt, and ground cinnamon to a large saucepan. Cook until the mixture reaches soft ball stage (236 degrees F). Pour the caramel over the popcorn and stir until all of the popcorn is coated.
Contextual Rewriter + GPT-2 Prompt	In a large pot, place your popped corn and cover it in the popped corn. Bring to a boil, stirring, then reduce the heat and simmer, stirring once or twice, for 20 minutes or until thick. Continue cooking for 5 minutes while gently stirring once in awhile to stop the edge of the pot from burning. Slowly pour in the corn syrup, and continue mixing until you can form a ball of dough. Roll out dough balls on a board lightly dusted with cornstarch to 1/4 to 1/2-inch thick.
Contextual Rewriter + Rule Prompt	Add nondairy butter, corn syrup and brown sugar to a medium saucepan over medium high heat. Combine the soymilk and dry ingredients in a medium bowl, then whisk in the wet. Cook over low heat, stirring constantly, about 10 minutes or until thickened; stir twice during cooking. Dump in the popcorn. Stir the mixture to coat it all with corn and pop it in the oven.

Table 13: Examples of document-level recipe rewrite outputs for Caramel Popcorn when the dietary constraint is dairy-free. Ingredients that violate the dietary constraint and their substitutions are colored.

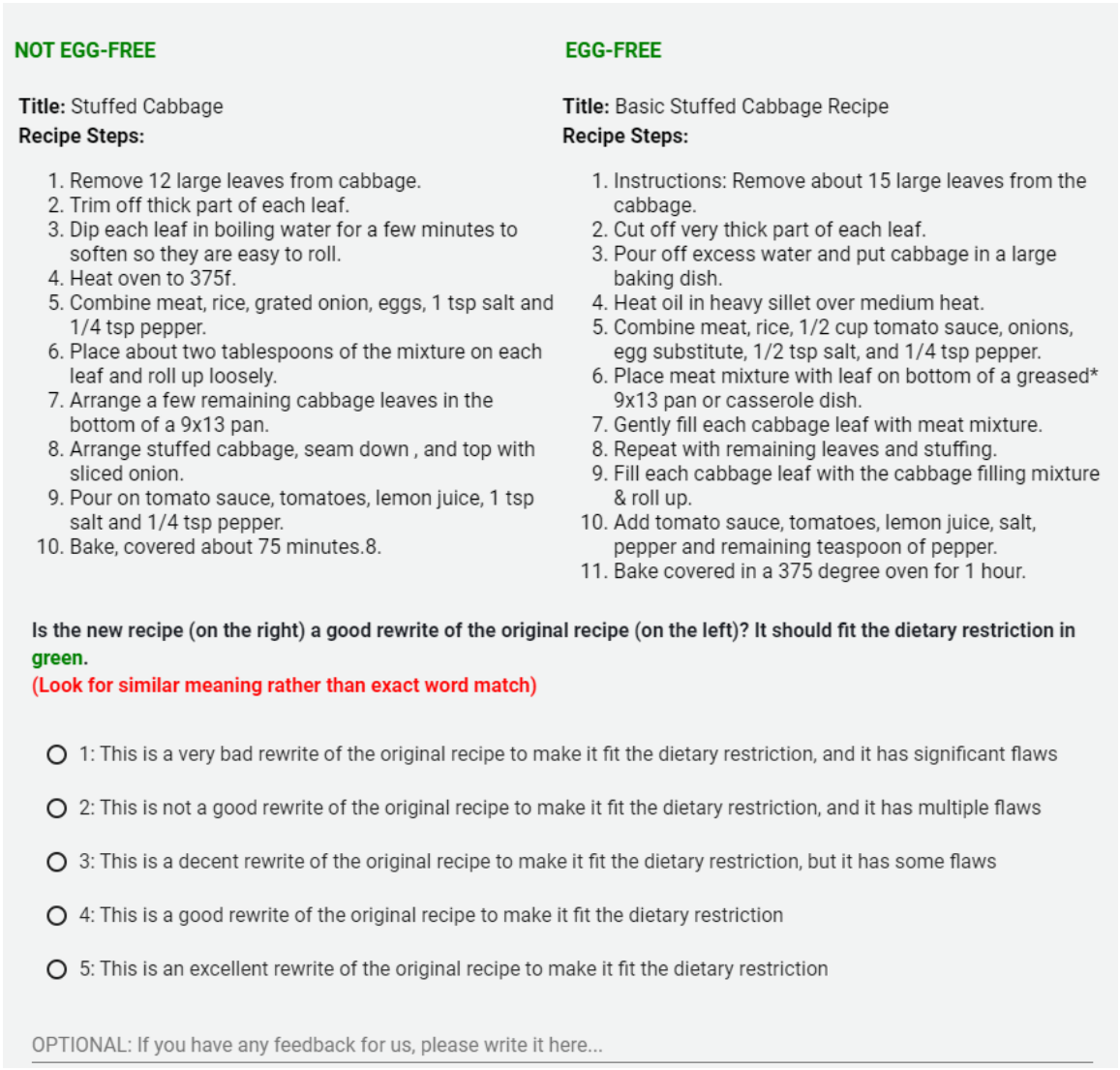

Figure 6: An example of a question used for human evaluation of the recipe rewrite task on Amazon Mechanical Turk.

Source Recipe Step	Model	Ingredient Prompt	Generated Recipe Step
a slice is delicious to eat on its own or with coconut butter, cashew butter, almond butter, or grass fed butter spread on top!	Contextual Rewriter + GPT-2 Prompt	banana, brown sugar, cinnamon, nutmeg, clove	In a large bowl, stir together banana, brown sugar, egg, butter and nutmeg.
	Contextual Rewriter + Rule Prompt	coconut, sunflower seed butter	Coat with your favorite healthy mix of seeds butter a few drops of liquid stevia or sweetener of choice sprinkled on top (optional).
Serve with milk or with a lump of butter melting into the nice warm pudding.	Contextual Rewriter + GPT-2 Prompt	spray oil, vanilla, chocolate bread pudding	Spray with cooking spray and add the chocolate bread pudding.
	Contextual Rewriter + Rule Prompt	soymilk, nondairy butter, pudding	Top pudding with dollops of vegan butter and a sprinkle of cinnamon.
Stir in beer.	Contextual Rewriter + GPT-2 Prompt	coconut, paprika	Heat up about 2 to 3 inches of coconut oil in a cast iron skillet.
	Contextual Rewriter + Rule Prompt	non-alcoholic beer	Slowly pour in 3/4 cup non-alcoholic beer, whisking briskly to blend.

Table 14: Examples to show the effect of the predicted and rule-based ingredient prompts on the generated steps.

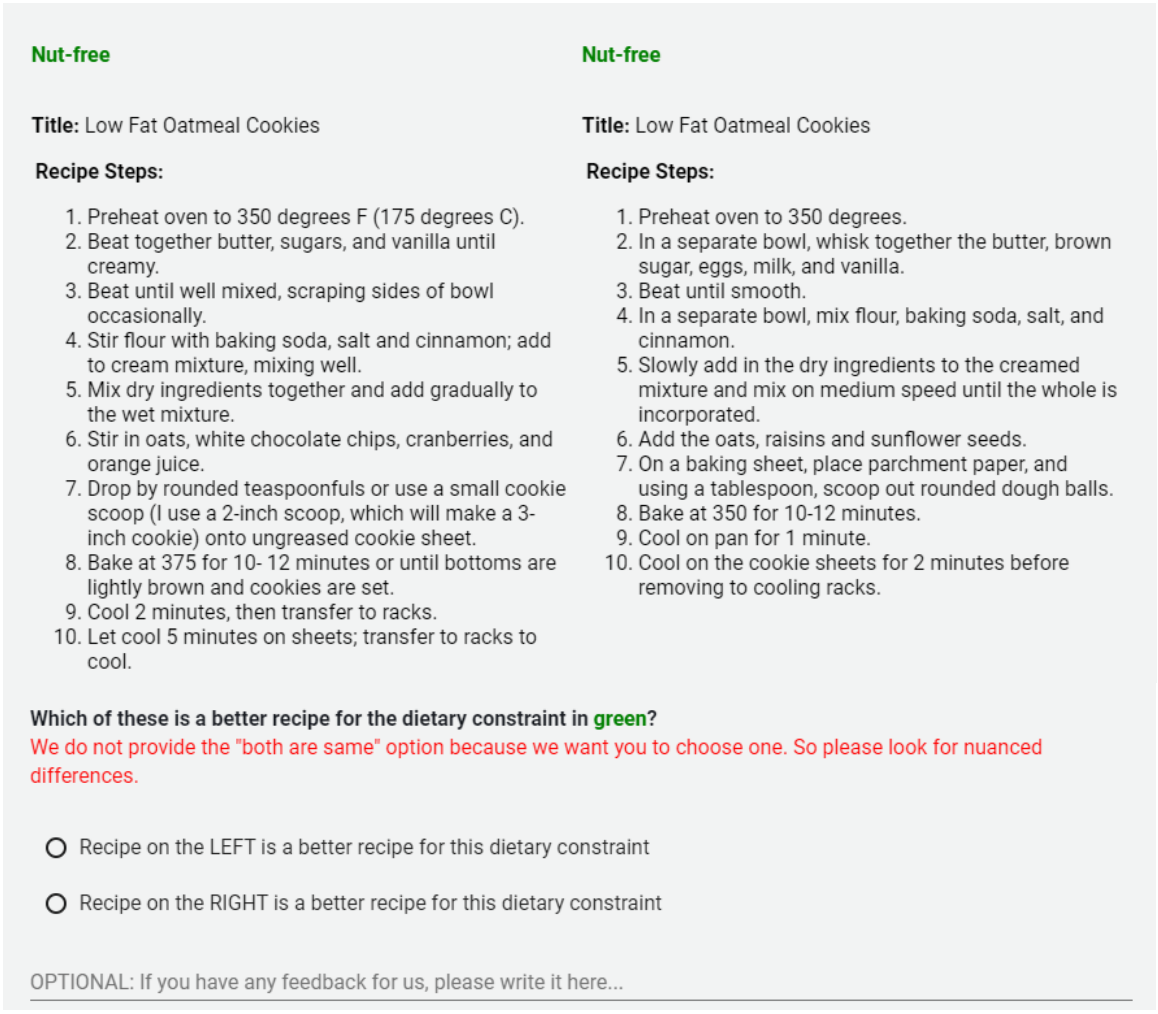

Figure 7: An example of a question used for human evaluation of the recipe rewrite task on Amazon Mechanical Turk.

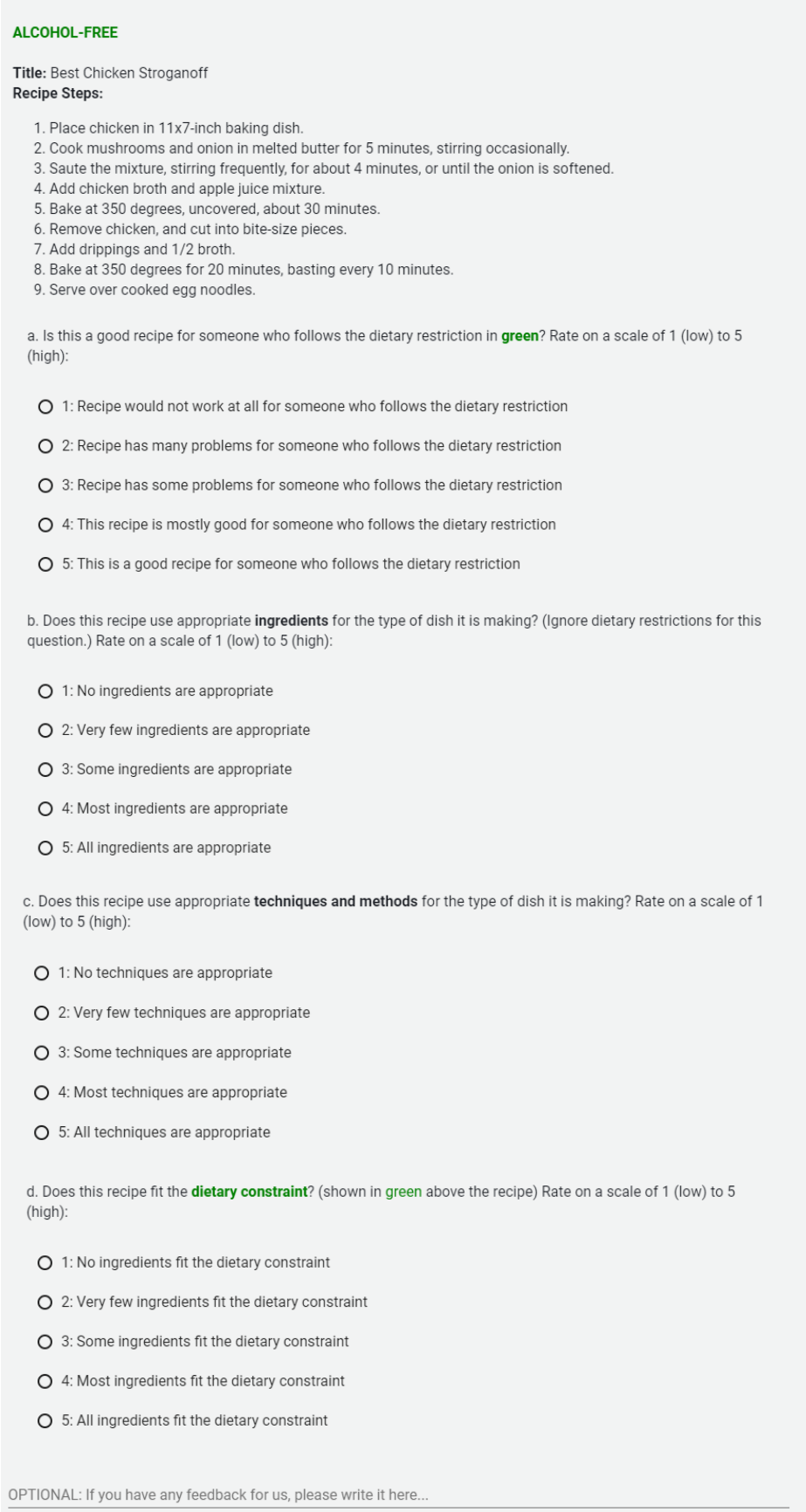

Figure 8: An example of a question used for human evaluation of the recipe rewrite task on Amazon Mechanical Turk.

Model	Fluency Perplexity ↓	Dietary Const. % Adherence ↑	Closeness to Source ROUGE ↑	Diversity Trigram ↑
Non-learning				
Rule-Based	10.92	96.6	*98.77*	0.557
Retrieval	11.09	93.9	26.78	0.380
Controllable Generation				
GPT-2	N/A	97.6	21.14	0.530
PPLM	12.85	95.2	20.83	0.577
CTRL	13.14	94.6	25.52	0.433
Sentence-level Transfer				
Seq2seq Copy	16.72	98.6	25.80	0.144
Transformer	9.85	96.3	27.54	0.328
Proposed Model Ablations				
End-to-End Rewriter	**9.69**	97.6	25.81	0.481
No-Context Rewriter	13.91	**99.9**	32.13	0.591
Contextual Rewriter	12.38	97.1	31.28	0.652
+ GPT-2 Ingredient Prompt	16.37	**99.8**	29.36	0.573
+ Rule Ingredient Prompt	14.60	**99.8**	**35.08**	**0.709**

Table 15: Automatic metric results on model rewrites of 1000 randomly sampled recipes from the dev set. The difference between bold and non-bold numbers is statistically significant with $p < 0.001$. We do not compare to *rule-based* under closeness to source since it copies steps from the source, leading to an artificially high score.

Model	Overall	Dairy	Nut-Free	Egg-Free	Vegan	Veget.	Alc.-Free	Fish-Free
Non-learning								
Rule-Based	96.1	95.1	96.9	96.5	93.5	98.6	98.9	97.8
Retrieval	93.4	91.9	99.2	95.5	84.9	92.8	96.4	98.8
Controllable Generation								
GPT-2	96.4	95.9	98.5	99.3	91.1	96.0	99.8	100.0
PPLM	94.9	92.9	97.6	99.5	89.1	93.6	100.0	100.0
CTRL	94.3	92.3	95.8	95.6	90.1	95.4	100.0	100.0
Sentence-level Transfer								
Seq2seq Copy	99.0	97.2	100.0	100.0	99.3	99.1	100.0	99.3
Transformer	93.5	89.8	98.1	98.7	87.5	92.2	98.7	100.0
Proposed Model Ablations								
End-to-End Rewriter	97.0	97.1	99.4	98.4	91.8	96.1	100.0	100.0
No-Context Rewriter	99.9	100.0	100.0	100	99.8	100.0	100.0	100.0
Contextual Rewriter	99.6	99.9	100.0	100.0	98.5	99.1	100.0	100.0
+ GPT-2 Ing. Prompt	99.6	99.7	99.7	99.6	98.9	99.5	100.0	100.0
+ Rule Ing. Prompt	99.5	99.7	99.7	100	99.2	98.2	100.0	99.2

Table 16: Further detail on dietary constraint accuracy for 1000 randomly sampled recipes from the test set.

Model	Fluency Perplexity ↓	Dietary Const. % Adherence ↑	Closeness to Source ROUGE ↑	Diversity Trigram ↑
Human Rewrite	13.91	99.7	77.08	0.906
Contextual Rewriter + Rule Ing. Prompt	20.28	96.3	35.44	0.836

Table 17: Comparison of the rewrites done by humans to the Contextual Rewriter + Rule Prompt on a subset of 30 vegetarian recipes from the test set.