

Grounded Complex Task Segmentation for Conversational Assistants

Anonymous ACL submission

Abstract

Following complex instructions in conversational assistants can be quite daunting due to the shorter attention and memory spans when compared to reading the same instructions. Hence, when conversational assistants walk users through the steps of complex tasks, there is a need to structure the task into manageable pieces of information of the right length and complexity. In this paper, we tackle the recipes domain and convert reading structured instructions into conversational structured ones. We annotated the structure of instructions according to a conversational scenario, which provided insights into what is expected in this setting. To computationally model the conversational step’s characteristics, we tested various Transformer-based architectures, showing that a token-based approach delivers the best results. A further user study showed that users tend to favor steps of manageable complexity and length, and that the proposed methodology can improve the original web-based instructional text. Specifically, 86% of the evaluated tasks were improved from a conversational suitability point of view.¹

1 Introduction

Voice-based assistants can guide users through everyday complex tasks, such as cooking, crafts, and home repairs. These conversational assistants need to understand the users’ intention, find a specific recipe, and communicate it in a structured and well-paced manner. Supporting this type of task-guiding interaction is a recent topic (Gottardi et al., 2022; Choi et al., 2022; Strathearn and Gkatzia, 2022), where conversational assistants must work hand-in-hand with users in order to guide them throughout the task execution.

We argue that most instructional texts found online are structured in a non-optimal way for conversational assistants, due to the inherent differ-

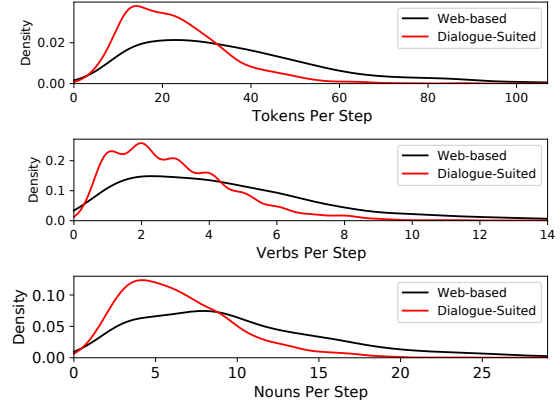


Figure 1: Differences between web-based and dialogue-suited recipes (i.e., ConvRecipes corpus) in terms of the density distribution of tokens, verbs, and nouns per step.

ences between screen and voice-based interfaces. Recipes are a great example where the decomposition of the recipe’s text into dialogue-suited steps is critical (example in Appendix A) – as Gottardi et al. (2022) observed, the user is dividing attention through various and possibly parallel actions. Hence, and following previous knowledge (Miller, 1956; Cowan, 2001), we aim for steps that are structured and presented to the user in ordered pieces of information, while dosing complexity, with the aim of achieving an efficient task completion.

To tackle this new problem, we part ways with topic-based segmentation methods (Koshorek et al., 2018; Choi, 2000; Arnold et al., 2019) and propose a novel human-focused methodology to convert reading-structured instructions into conversational ones. Figure 1 offers a clear view of the differences between the original web-based recipes and their dialogue-suited counterparts. The distribution of linguistic characteristics such as length, verbs (which cover actions), and nouns (covering ingredients, tools, etc.), confirms that dialogue-suited instructions should avoid overwhelming users’ short-term memory (Miller, 1956; Cowan, 2001).

Our proposed methodology starts with the cura-

¹The dataset and code will be made publicly available.

tion of a corpus which we call *ConvRecipes*, where online recipes are segmented into recipes with steps more suited to a conversational agent. Moreover, we identify the key traits of a conversational step. An example of this can be seen in Figure 2. This example shows the need for models that can tackle our task, in specific, in "Step 2" of the task, where it is noticeable that the step would be very difficult to follow in a conversational assistant due to the long sentence and the inherent complexity of its actions.

To tackle this task computationally, we propose the Dialogue-Task Segmenter Transformer (*DTS-Transformer*), which follows state-of-the-art approaches in text-segmentation (Lukasik et al., 2020; Lo et al., 2021; Solbiati et al., 2021) and adopts a Transformer-based backbone (Vaswani et al., 2017). Distinct from previous work, we follow a token-level approach which by modeling steps' text at a finer granularity, is capable of better modeling the inherent structural characteristics of conversational tasks. Note that, we did not follow generative approaches and ground our task segmentation task on the recipes' original text. We do this to avoid the risk of introducing hallucinations or mistakes in step-by-step procedures (Choi et al., 2022).

Finally, we validated the proposed methodology with automatic experiments, and, more interestingly, with a user evaluation. We observed that the best *DTS* model, a *T5-3B* Encoder backbone, trained on the proposed *ConvRecipes* corpus, was able to improve the conversational structure of 86% of the evaluated tasks. This evidences both the conversational characteristics of the *ConvRecipes* corpus and the effectiveness of the model's approach to the grounded conversational task segmentation task.

Next, we will relate our contributions to previous corpus and methods. In Section 3, we carefully detail the proposed methodology. Experimental validation and user evaluations are presented in Section 4, and we conclude with the final takeaways and future work.

2 Related Work

Related Corpora. While conversational-suited task segmentation is a novel task, multiple datasets have been created to address article-based text segmentation, with the earliest ones being the Choi Dataset (Choi, 2000), where each document is represented by the concatenation of 10 random

Title: Baked Bananas Recipe

Web-based Recipe

Step 1: Preheat oven to 190 degrees C. Spray a baking dish with cooking spray.

Step 2: Arrange banana halves in the prepared baking dish.

Drizzle maple syrup over bananas and top with cinnamon.

Bake in the oven until heated through, 10-15 minutes.

Dialogue-suited Recipe

Step 1: Preheat oven to 190 degrees C.

Step 2: Spray a baking dish with cooking spray.

Step 3: Arrange banana halves in the prepared baking dish.

Drizzle maple syrup over bananas and top with cinnamon.

Step 4: Bake in the oven until heated through 10-15 minutes.

Figure 2: Example of conversion from web/reading-based format to a dialogue-suited format. In blue and orange, we highlight the verbs and nouns, respectively.

passages from a large corpus, and the RST-DT dataset (Carlson et al., 2001), which focuses on intra-sentence granularity on Wall Street Journal articles. Topic and document-section-oriented segmentation datasets such as Wiki-727 (Koshorek et al., 2018) and WikiSection (Arnold et al., 2019) are comprised of Wikipedia articles and focus on topic and section-based text segmentation. Closer to our domain, we highlight works with instructional text such as Task2Dial (Strathearn and Gkatzia, 2022) and the Wizard of Tasks (Choi et al., 2022), which rewrite the tasks' text into dialogue-suited steps. Our approach focuses on grounded structuring of task instructions for dialog while avoiding hallucination problems common in generative/re-writing approaches. We also take a step further by identifying the fundamental traits of conversational-suited tasks, in a principled manner.

Methods and Models. Initial works for text segmentation were mostly based on statistical and unsupervised approaches, such as TextTiling (Hearst, 1997) and C99 (Choi, 2000). After these, supervised neural methods, particularly with the use of RNNs were utilized. In (Badjatiya et al., 2018), a CNN is used to generate sentence embeddings in conjunction with an LSTM to keep sequential information. Li et al. (2018) also presents an RNN-based model with an additional pointing mechanism and in (Koshorek et al., 2018) it is used a hierarchical Bi-LSTM model.

Currently, the state-of-the-art is based on supervised Transformer-based approaches (Lukasik et al., 2020; Lo et al., 2021; Solbiati et al., 2021). In (Lukasik et al., 2020), cross-segment and hierarchical models are proposed, where predictions are made based on consecutive segments or sentence-

based representations of the segments. Lo et al. (2021) presented a hierarchical approach combining sentence and cross-segment embeddings. Xing et al. (2020) proposed a hierarchical BiLSTM to complement BERT’s (Devlin et al., 2019) sentence representations, aided by a coherence-related auxiliary task. Some approaches such as (Zhang et al., 2021), tackle task structuring as a generation task, where an end-to-end pipeline is proposed to generate day-to-day tasks. In a dialogue setting, Solbiati et al. (2021) applied a BERT model for transcript-based meetings segmentation (Janin et al., 2003; McCowan et al., 2005) by calculating the similarity between segment embeddings given by a pre-trained model. Given the particular intricacies of conversational-suited task structuring, while we also adopt a Transformer backbone, we propose a task segmentation model that makes decisions at a token-level being able to consider the global task’s structure.

3 Structuring Conversational Tasks

Our hypothesis is that the recipe instructions found online are not suited for conversational assistants, motivating both the task and the dataset collection efforts. To convert instructions from a reading structure into a conversationally structured format, we followed a human-focused methodology. First, we collected task instructions and ran a user study to curate them as conversational instructions. Second, we ask users to annotate the relevance of various conversational instructions traits. Third, we analyzed the linguistic characteristics of reading instructions compared to conversational task instructions. Finally, we modeled conversational-steps computationally with various Transformer-based (Vaswani et al., 2017) architectures.

3.1 A Conversational-Tasks Corpus

Currently, there are no explicit corpora for studying the grounded segmentation of a recipe into conversational-suited steps. The closest examples are either section-based document segmentation (Koshorek et al., 2018; Arnold et al., 2019) or rewriting/generative approaches (Choi et al., 2022; Strathearn and Gkatzia, 2022) which are prone to hallucinations. In this section, we introduce the methodology used to create the *ConvRecipes* corpus, consisting of recipes segmented into conversational-suited steps.

3.1.1 Tasks Collection and Annotation

To create the *ConvRecipes* corpus, we collected recipes from a popular recipes website, where each recipe is self-contained and composed of various steps in English with arbitrary lengths. We started by filtering out recipes with fewer than three steps due to having a structure that is too simple. After this, near-duplicate recipes were identified with SimHash (Charikar, 2002) and removed.

Conversation-Steps Annotation. Even though recipes are human-edited, we argue that they are written for a reading-based setting, making them ill-suited to be used in a conversational setting. Hence, to create grounded conversational instructions, we conducted a user study. In total, we had 8 annotators, 6 male and 2 female all Computer Science MSc. and or Ph.D. students. All annotators had experience with both conversational assistants and cooking applications, making them particularly suited for this annotation task.

The annotators were shown the original recipes and asked to propose (or not) changes to make the recipes dialog-suited, either by adding and/or removing steps. Figure 2 illustrates the annotation process: given a recipe formatted for the Web, the goal is for the annotator to identify the structure that is better suited for a conversational setting. This approach makes the segmentation grounded on the original task, avoiding the introduction of mistakes prone to happen when using rewriting approaches.

3.1.2 On the Traits of a Conversational Step

After the annotation process described in the previous section, the annotators were asked to quantify, on a Likert scale of 1 to 5, the importance of various conversational traits. In particular, we considered: *Complexity*, *Clarity*, *Length* and *#Steps*, *Ability to Parallelize Tasks*, and *Naturalness*. For the exact description of these traits refer to Appendix B. This evaluation of the traits aims to further inform us what users value in this conversational task-guiding assistance setting (Gottardi et al., 2022).

Table 1 shows the results of the analysis of the traits. The results reveal that although all traits have some importance, users mostly focus on the complexity and length of the steps, which are generally connected with each other. This means that managing complexity and ensuring a balance in the information given to the user is paramount. On the other hand, the naturalness and the ability to per-

Conversational-Step Trait	Importance
(1) Complexity	4.5
(2) Step Length & #Steps	4.2
(3) Clarity	3.8
(4) Naturalness	3.6
(5) Ability to Parallelize Tasks	3.4

Table 1: Trait importance on a 1 to 5 scale. A higher value represents higher importance.

	Reading	Dialog
Avg. # Tokens	135	
Avg. # Sentences	9.3	
Avg. # Steps	3.80	5.85
Avg. # Tokens step	35.44	23.03
Avg. # Sents. step	2.44	1.59
Avg. # Verbs step	4.23	2.75
Avg. # Nouns step	9.92	6.44

Table 2: Comparison between the 300 original reading-based recipes and the manually annotated set.

form parallel tasks were considered less important traits, which seem to indicate that users are not so concerned with language naturalness given that the step should be short and not too complex.

3.2 ConvRecipes Corpus Analysis

After preparing and curating the task instructions, we analyzed and compared the original to the curated data in order to understand how the language differs from a web/reading setting to a conversational setting.

3.2.1 Reading-suited vs Dialog-suited

In total, 300 recipes were annotated, where 59 recipes were left without changes, and the remaining 241 (80.3%), had at least one new step added, with one, two, and three or more breaks added 75, 66, and 47 times, respectively. Only one recipe was annotated with fewer steps than the original. This result shows that the reading-based instructions are not optimal for a conversational setting, generally missing critical segmentations. Table 2 further evidences the difference between the original and the conversational-suited instructions, where it is clear that there is a preference for shorter segments with fewer actions. These results correlate with the importance of the conversational traits (Table 1), which showed that the complexity and number of steps are particularly important in this setting.

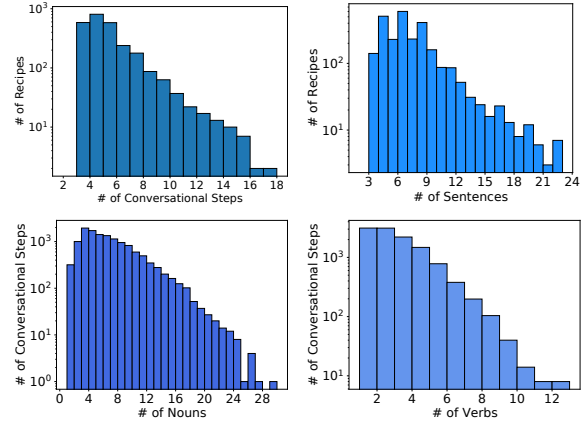


Figure 3: ConvRecipes statistics: conversational-steps per task (top-left), sentences per task (top-right), nouns per conversational-step (bottom-left), and verbs (bottom-right) per conversational-step.

Thus, ConvRecipes presents a step forward in discovering the optimal structure for instructional text in a conversational scenario.

3.2.2 Linguistic Style of Conversational-Steps

Figure 3 shows the corpus’s distribution of conversational steps, sentences, nouns, and verbs. The figure indicates that there is a lot of variability that needs to be correctly addressed, due to each recipe having a particular structure.

In contrast to other corpora (Koshorek et al., 2018; Choi, 2000), ConvRecipes is written in an instructional/imperative format, using actionable verbs mostly related to the cooking domain such as “stir”, “bake” and “mix”. Analyzing how steps start and end, can also bring some insights into the segmentation behavior, so we examined the most common starting and ending n-grams of each step. The top-20 starting and ending tri-grams are available in Figure 4. This showed that many of the steps have temperature mentions, e.g. “preheat oven to”, or time aware mentions e.g., “for [N] minutes” ([N] is a placeholder replacing the number). These indicate a start/end of an action which in turn reflects a new step. It is important to note that the majority of both bi-grams (65%) and tri-grams (80%) are only used once, which shows the diversity of actions available, creating a more complex challenge for data-driven approaches.

3.2.3 Corpus Processing.

Annotating a large number of recipes is labor-intensive and expensive. Thus, we use the 300 manually annotated recipes as the test set and cre-

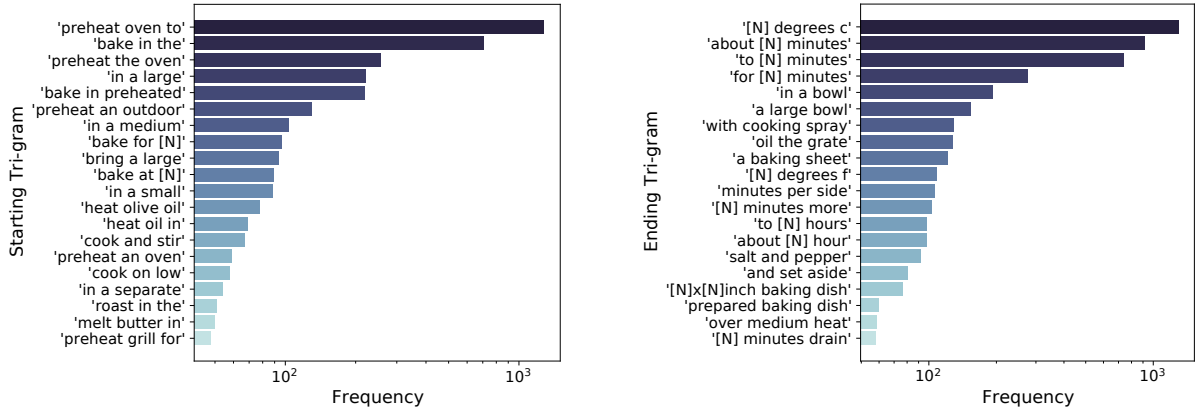


Figure 4: Distribution of the top-20 most frequent starting (left) and ending (right) tri-grams.

ate the training and validation splits automatically, by using the average number of sentences per step of the annotated set (1.59, Table 2) as a maximum threshold to choose recipes from a non-annotated set. These non-annotated recipes use the original step information as the ground-truth step labels, in a similar way as the section markers in the Wiki-727 dataset (Koshorek et al., 2018). This resulted in a dataset with 1930, and 424 recipes for training and validation, respectively. As mentioned before, the test set is composed of the 300 annotated recipes.

To conclude, the aim of this corpus is to create models that learn how to identify steps and segment a task into grounded dialogue-suited steps. Hence, we concatenate all the steps together, resulting in an unstructured text with no segment-identifying structure. Step annotations are then used as labels to train and evaluate the models.

3.3 Dialog-Task Structuring Transformer

To learn the structure of a conversational task, we processed the entire task’s text as a whole. By explicitly receiving the entire input sequence, we aim to take into account the size and position of each segment token relative to all the other tokens. Consequently, with a single pass over the input, this approach is able to output all segment predictions, making it more efficient than sentence-based embedding models that output a prediction per sentence (Lukasik et al., 2020).

Given the characteristics of the Transformer model (Vaswani et al., 2017), we use it as the basis for our *Dialogue-Task Structuring* (DTS) model. In particular, we feed the model with the complete recipe, allowing it to create contextualized token representations of the entire recipe. This allows the

model to consider all of the tokens in the recipe via the self-attention mechanism.

After the input has been processed by the Transformer, we apply a binary segment-break prediction head, i.e., a linear layer followed by a *softmax* to the embedding of each segment identifying token (emb_t), outputting the probability of a token (t) being a *segmentation token*:

$$P_{seg}(t_i) = softmax(FFNN(emb_{[t_i]})), \quad (1)$$

To identify these end-of-segment tokens, generally punctuation marks (e.g. “.”, “!”, “;”), we use Spacy (Honnibal and Johnson, 2015) to perform basic sentence segmentation over the recipe’s text. Finally, we train the model using the cross-entropy loss between the model predictions, \hat{y} , and the binary segmentation labels, y , as the following:

$$L_{CE} = y \cdot \log \hat{y} + (1 - y) \cdot \log (1 - \hat{y}), \quad (2)$$

4 Experiments

In this section, we demonstrate how the proposed framework tackles the challenge of structuring task instructions in a conversational setting. Experimental validation was done with both automatic metrics and human evaluation.

4.1 Metrics

We use Precision, Recall, and F-score to measure the detection of the correct location of a conversation step following Li et al. (2018). Moreover, we followed (Koshorek et al., 2018; Arnold et al., 2019), and used the text segmentation metric P_k (Beeferman et al., 1999), which compares the predicted segmentation with the ground-truth labels, where a lower value represents a better model.

4.2 DTS Backbones and Implementation Details

As the basis for our models, we used pre-trained Transformer models. We tested with the encoder-only model BERT (Devlin et al., 2019), the encoder-decoder model T5 (Raffel et al., 2020) in both an encoder-decoder (Enc-Dec) setting and in an encoder-only (Enc-only), i.e. decoder is not used. When using an E-D model, the input sequence of the decoder is the same as the encoder as in (Lewis et al., 2020) for the extractive QA task (i.e. there is no actual decoding). To identify the candidate segments, we used Spacy (Honnibal and Johnson, 2015), to be more robust than a simple punctuation-based approach. We evaluated in the test set the model with the best performance in the validation set in terms of F-Score. Additional information about model training is provided in Appendix C.

4.3 Baselines

As baselines, we considered random and uniform approaches, a classic method (Hearst, 1997), and a strong baseline based on a cross-encoder (Lukasik et al., 2020):

Rand_p and **Every_n**: unsupervised methods which use Spacy (Honnibal and Johnson, 2015) to identify sentences. p is the probability of breaking at each sentence, and n is the number of consecutive sentences to break.

TextTiling (Hearst, 1997): one of the earliest text segmentation methods based on lexical co-occurrence.

Cross-Segment (CrossSeg) (Lukasik et al., 2020): BERT-Base (Devlin et al., 2019) model with a classification head that predicts if a pair of input sentences should be segmented.

4.4 Results and Discussion

In Table 3, we present the results of the baselines, along with the results of the proposed DTS models.

Importance of Conversational-Aware Corpora. We trained the same DTS model with a BERT-Base backbone: one on all crawled raw recipes (20.000 recipes), identified as (*All**), and one on the ConvRecipes training set (*BERT-Base*). The results on Table 3, show that *BERT-Based (All*)* obtained the highest Precision (93.4), since its training samples have fewer breaks, the model makes less, but correct, break predictions. On the other

hand, it achieved the lowest Recall of all supervised methods. More importantly, the results clearly show the importance of training models with suited data, yielding a P_k relative improvement of 15% (*Bert-Base*). This result indicates that the ConvRecipes dataset is constructed in a way that embeds the traits of conversational task instructions (Section 3.1.2).

General results. In Table 3, we observe that the baselines *Rand_p*, *Every_n* and *TextTiling* do not generally break the steps at the correct locations as indicated by their low precision ($\leq 62\%$). However, since it implicitly enforces a step distribution that resembles the dataset, *Every₁* achieves a fairly good P_k , while also achieving a recall close to 100% due to breaking at every sentence (it is $\neq 100\%$ due to errors in Spacy’s sentence identification algorithm). *TextTiling*, which decides the task structure through lexical overlap, performs poorly and does not appear to be a good option for this task. This is because recipe steps are not structured based on overlap, but rather in a sequence of sub-actions, which *TextTiling* overlooks.

The *CrossSeg* achieved a P_k of 19.5 which is already a significant improvement over the best unsupervised baseline which achieved 23.3. This translates into an F1 score improvement from 73.8% to 76.5%. Regarding the *DTS* models, the most solid fact that emerges from Table 3 is that, regardless of the backbone, our *DTS* approach consistently outperforms all the baselines.

In general, the results of the baselines illustrate the difficulty of the problem we are trying to solve. Moreover, there is a clear divide in terms of P_k between previous baselines and the proposed *DTS* framework, which is consistently below 20 P_k , highlighting the importance of capturing the relations between the task and the conversational steps.

Encoder vs Encoder-Decoder Backbones.

Comparing the encoder-only model BERT (Devlin et al., 2019) with T5 (Raffel et al., 2020) (Enc-only) or the full encoder-decoder (Enc-Dec) model, in situations with a comparable number of parameters, we see that *T5* outperforms *BERT*. This might be explained by the different pre-training approaches used in T5 (Raffel et al., 2020), which are better suited for our task. Comparing the encoder-only (Enc-only) with the encoder-decoder (Enc-Dec) in the same models, we see an improvement in *T5-Large*, but a decrease in performance in

	Model	# Params	$P_k \downarrow$	Precision \uparrow	Recall \uparrow	F1 \uparrow
Baselines	Rand _{0.5}	-	35.4 \pm 0.3	59.9 \pm 0.5	49.7 \pm 0.8	51.7 \pm 0.6
	Rand _{0.75}	-	28.3 \pm 0.5	61.2 \pm 0.4	75.0 \pm 0.9	65.2 \pm 0.6
	Every ₁	-	23.3	60.9	98.8	73.8
	Every ₂	-	37.9	59.6	37.9	44.9
	TextTiling	-	28.4	58.7	67.7	61.4
	CrossSeg	110 M	19.5 \pm 0.4	77.5 \pm 0.9	79.5 \pm 1.6	76.5 \pm 0.4
Dialogue Task Segmenter (DTS)	BERT-Base (All*)	110 M	22.5 \pm 0.3	93.4 \pm 0.1	58.7 \pm 0.4	69.6 \pm 0.4
	BERT-Base	110 M	19.1 \pm 0.4	75.8 \pm 0.7	83.6 \pm 0.7	77.5 \pm 0.4
	BERT-Large	340 M	18.4 \pm 0.2	77.0 \pm 1.7	83.6 \pm 2.8	78.1 \pm 0.5
	T5-Base (Enc-only)	110 M	17.7 \pm 0.2	77.9 \pm 0.7	84.2 \pm 0.5	79.0 \pm 0.1
	T5-Base (Enc-Dec)	220 M	18.1 \pm 0.6	77.9 \pm 0.3	82.9 \pm 1.6	78.5 \pm 0.8
	T5-Large (Enc-only)	335 M	18.1 \pm 0.2	77.4 \pm 0.4	84.1 \pm 0.4	78.6 \pm 0.3
	T5-Large (Enc-Dec)	770 M	17.7 \pm 0.2	79.1 \pm 0.8	81.9 \pm 0.9	78.5 \pm 0.2
	T5-3B (Enc-only)	1.5 B	17.0 \pm 0.4	78.3 \pm 1.0	85.9 \pm 0.9	80.0 \pm 0.2

Table 3: Results on the ConvRecipes’s test set from an average of 3 runs per model. All* indicates that the model was trained on the set of all recipes crawled, in their original form.

T5-Base. This result implies that the use of the decoder part of *T5* might not be necessarily needed for this particular task.

DTS Model Size Influence. Having established the performance range of *DTS*, we examined the relationship between model size and performance. Results show that increasing the model size can bring improvements, in particular, from *BERT-Base* to *BERT-Large*, however, in the case of *T5-Base* for *T5-Large*, we notice an improvement in the Enc-Dec model and a decrease in performance in the encoder-only (Enc-only) model. Nonetheless, the best results by a significant margin in P_k and F1 are obtained with the largest model *T5-3B* (Enc-only), showing that the use of larger models can bring an improvement as evidenced in (Raffel et al., 2020; Chowdhery et al., 2022).

Main Takeaways. Results indicate that the proposed models are capable of capturing the intrinsic relations of the steps and extract them correctly when trained on high-quality conversationally structured task instructions. We also observe that our *DTS-Transformer* approach gives the best results in this setting. A fundamental difference between *DTS* and other supervised approaches is that it tackles the conversational recipe structuring task at a token-level granularity. As a consequence, it abstracts less information than previous approaches (Lukasik et al., 2020; Lo et al., 2021), such that at each Transformer layer, intermediate token embeddings are contextualized on the full-task sequence. Despite working at a finer granularity (token-level), *DTS* is both faster to train and perform inferences. This makes it highly suited to

be applied in a real setting, to structure tasks into conversational steps.

4.4.1 Conversational Tasks Statistics

The recipe task structuring results led us to further examine the resulting conversational steps statistics. These are shown in Table 4, where we contrast the # Steps and # Tokens statistics with the human-annotated set. Specifically, we observe that Exact Match segmentation is higher (17.0%) in the *DTS T5-3B (Enc-only)* model due to its greater ability to capture the segmentation patterns. It is also interesting to note that all methods have a tendency to overestimate the number of steps. Finally, for $\Delta\text{Steps} \leq 1$ – the percentage of examples where the model predicts less than one step of difference with the test set – we see an equivalent performance within the supervised baselines.

Overall, by examining these task structuring statistics, we observe that although the average number of steps (# Steps column) is acceptable for most methods, when we look at the finer-grain statistics, we see that there is a non-trivial balance between step length, number of steps, and content of each step. Hence, it is not a sufficient condition to optimize a single statistic but rather a combination of these.

4.4.2 User Evaluation

To compare the model’s performance to the original web-based instructions, we asked 6 annotators from the same pool of Section 3 to annotate which segmentation was the best considering a conversational setting.

In total, 50 recipes were randomly selected from

		# Steps	# Tokens	Exact Match	= # Steps	+ # Steps	- # Steps	$\Delta\text{Steps} \leq 1$
	Human Annotation	5.86	19.21	-	-	-	-	-
Method	Every ₁	9.29	12.11	5.00%	5.33%	94.67%	0.00%	24.00%
	Text Tiling	6.32	17.80	7.00%	24.00%	49.33%	26.67%	58.67%
	CrossSeg	6.08	18.53	13.33%	30.67%	36.22%	33.11%	68.11%
	DTS T5-3B (Enc-only)	6.48	17.37	17.00%	27.56%	46.44%	26.00%	68.44%

Table 4: Detailed conversational task structuring statistics for the ConvRecipes test set (human annotated). Exact Matches is the percentage of predictions exactly matching the ground-truth. (=, + and -) # Steps represent the percentage of predictions that have equal, more, or less steps than the ground-truth. $\Delta\text{Steps} \leq 1$ indicates the percentage of times the difference between the # Steps predicted and the ground-truth is ≤ 1 step.

	Web-based	T5-3B (E-only)
Rating 1	18.0%	3.3%
Rating 2	36.0%	12.7%
Rating 3	18.7%	20.7%
Rating 4	20.0%	35.3%
Rating 5	7.3%	28.0%
Best	14.0%	86.0%
Conv. Suitability	2.63	3.72

Table 5: User study results comparing the original web-based segmentations with T5-3B (Enc-only) model predictions. (Conversation Suitability is given on a 1 to 5 scale.

the test set in their original web-based format (i.e., without human annotations). These recipes were then compared to the predictions of the best model *DTS T5-3B* (Enc-only). Examples can be seen in Appendix D. For each recipe, we collected 3 annotations, resulting in an inter-rater agreement of 73% w.r.t. binary preference. Additionally, the annotators were also asked to grade each segmentation (web and model) on a 1 to 5 Likert scale according to the suitability for a conversational agent.

Table 5 shows the results of the user evaluation. We observe a preference for the model’s segmentation (86%) since it was trained on a conversational-based data distribution which more accurately reflects the user’s preference in this setting. We also analyzed that the annotators had a preference for recipes with more segments 88% of the time. Notwithstanding, it is important to note that breaking too often may result in a sub-optimal experience and in incomplete steps, as shown by the *Every₁* baseline of Table 3.

Considering the 1 to 5 rating of suitability for a conversational agent, the model’s prediction scores were much higher (3.72) than the original recipes

(2.63). These ratings further reinforce our hypothesis that the original recipes are not dialogue suited, and that the model is able to greatly increase the suitability of a recipe to a conversational-friendly format.

To conclude, these results indicate that the model is able to capture segmentation patterns, showing an ability to improve the suitability of a recipe for a conversational assistant. This, in turn, brings advantages to the user experience by providing a grounded conversationally-suited segmentation.

5 Conclusions

In this paper, we proposed a methodology to tackle the problem of converting web/reading structured instructions into conversationally structured ones. In summary, the key contributions are as follows:

ConvRecipes Corpus. This corpus enables a better understanding of the problem. Its analysis showed that instructional text as it is presented online is not optimal for a conversational setting.

Dialogue-Task Structurer (DTS). We proposed several methods that can effectively capture segmentation linguistic patterns. The best-performing method was a T5-3B (Enc-only) model, a token-level Transformer.

Real-World Improvement. The user evaluation showcased the model’s ability to improve over the original segmentation (86%), which brings advantages in user experience in a conversational-assistant scenario.

For future work, we intend to assess how segmentation influences the user’s perception of a recipe’s quality and generalize our experiments to different domains such as DIY tasks and tutorials.

References

- Sebastian Arnold, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A. Gers, and Alexander Löser. 2019. [SECTOR: A neural model for coherent topic segmentation and classification](#). *Trans. Assoc. Comput. Linguistics*, 7:169–184.
- Pinkesh Badjatiya, Litton J. Kurisinkel, Manish Gupta, and Vasudeva Varma. 2018. [Attention-based neural text segmentation](#). In *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings*, volume 10772 of *Lecture Notes in Computer Science*, pages 180–193. Springer.
- Doug Beeferman, Adam L. Berger, and John D. Lafferty. 1999. [Statistical models for text segmentation](#). *Mach. Learn.*, 34(1-3):177–210.
- Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. [Building a discourse-tagged corpus in the framework of rhetorical structure theory](#). In *Proceedings of the SIGDIAL 2001 Workshop, The 2nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, Saturday, September 1, 2001 to Sunday, September 2, 2001, Aalborg, Denmark*. The Association for Computer Linguistics.
- Moses Charikar. 2002. [Similarity estimation techniques from rounding algorithms](#). In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 380–388. ACM.
- Freddy Y. Y. Choi. 2000. [Advances in domain independent linear text segmentation](#). In *6th Applied Natural Language Processing Conference, ANLP 2000, Seattle, Washington, USA, April 29 - May 4, 2000*, pages 26–33. ACL.
- Jason Ingyu Choi, Saar Kuzi, Nikhita Vedula, Jie Zhao, Giuseppe Castellucci, Marcus Collins, Shervin Malmasi, Oleg Rokhlenko, and Eugene Agichtein. 2022. [Wizard of tasks: A novel conversational dataset for solving real-world tasks in conversational settings](#). In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 3514–3529. International Committee on Computational Linguistics.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. [Palm: Scaling language modeling with pathways](#). *CoRR*, abs/2204.02311.
- Nelson Cowan. 2001. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and brain sciences*, 24(1):87–114.
- 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Anna Gottardi, Osman Ipek, Giuseppe Castellucci, Shui Hu, Lavina Vaz, Yao Lu, Anju Khatri, Anjali Chadha, Desheng Zhang, Sattvik Sahai, Preerna Dwivedi, Hangjie Shi, Lucy Hu, Andy Huang, Luke Dai, Bofei Yang, Varun Somani, Pankaj Rajan, Ron Rezac, Michael Johnston, Savanna Stiff, Leslie Ball, David Carmel, Yang Liu, Dilek Hakkani-Tur, Oleg Rokhlenko, Kate Bland, Eugene Agichtein, Reza Ghanadan, and Yoelle Maarek. 2022. [Alexa, let’s work together: Introducing the first alexa prize taskbot challenge on conversational task assistance](#). In *Alexa Prize TaskBot Challenge Proceedings*.
- Marti A. Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguistics*, 23(1):33–64.
- Matthew Honnibal and Mark Johnson. 2015. [An improved non-monotonic transition system for dependency parsing](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.
- Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. 2003. [The ICSI meeting corpus](#). In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP ’03, Hong Kong, April 6-10, 2003*, pages 364–367. IEEE.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

- Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. [Text segmentation as a supervised learning task](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 469–473. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.
- Jing Li, Aixin Sun, and Shafiq R. Joty. 2018. [Segbot: A generic neural text segmentation model with pointer network](#). In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4166–4172. ijcai.org.
- Kelvin Lo, Yuan Jin, Weicong Tan, Ming Liu, Lan Du, and Wray L. Buntine. 2021. [Transformer over pre-trained transformer for neural text segmentation with enhanced topic coherence](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3334–3340. Association for Computational Linguistics.
- Michal Lukasik, Boris Dadachev, Kishore Papineni, and Gonalo Simões. 2020. [Text segmentation by cross segment attention](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4707–4716. Association for Computational Linguistics.
- Iain McCowan, Jean Carletta, Wessel Kraaij, Simone Ashby, S Bourban, M Flynn, M Guillemot, Thomas Hain, J Kadlec, Vasilis Karaiskos, et al. 2005. The ami meeting corpus. In *Proceedings of the 5th international conference on methods and techniques in behavioral research*, volume 88, page 100. Citeseer.
- George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *J. Mach. Learn. Res.*, 21:140:1–140:67.
- Alessandro Solbiati, Kevin Heffernan, Georgios Damaskinos, Shivani Poddar, Shubham Modi, and Jacques Calì. 2021. [Unsupervised topic segmentation of meetings with BERT embeddings](#). *CoRR*, abs/2106.12978.
- Carl Strathearn and Dimitra Gkatzia. 2022. [Task2dial: A novel task and dataset for commonsense-enhanced task-based dialogue grounded in documents](#). In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering, DialDoc@ACL 2022, Dublin, Ireland, May 26, 2022*, pages 187–196. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Linzi Xing, Brad Hackinen, Giuseppe Carenini, and Francesco Trebbi. 2020. [Improving context modeling in neural topic segmentation](#). In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pages 626–636. Association for Computational Linguistics.
- Yi Zhang, Sujay Kumar Jauhar, Julia Kiseleva, Ryen White, and Dan Roth. 2021. [Learning to decompose and organize complex tasks](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2726–2735. Association for Computational Linguistics.

A Task Grounded Dialogue Segmentation Example in Conversation

In Table 6, we show an example dialog between a user and a conversational assistant. It is clear that segmenting a task is critical to managing the user’s cognitive load, and allows for a better understanding of the recipe’s multiple steps.

Table 6: Example of a dialogue-suited segmented recipe in the context of a conversational task assistant.

System	User
Hello! I am your cooking assistant! Try asking me things like: How to make lasagna? or How to bake a cake?	I want to make a Tiramisu.
Sure, I found this very tasty: "Classic Tiramisu". Do you want to try it?	Sure.
Step 1. Separate eggs. Whisk egg yolks with confectioner’s sugar until pale and fluffy.	How many eggs should I use?
This recipe uses 2 eggs.	Next.
Step 2. Sure, let’s keep going. Add Amaretto and mascarpone. Whisk until smooth.	Next Step.
Step 3. Beat egg whites with a pinch of salt until stiff peaks form. Slowly whisk in the remaining confectioner’s sugar.	Go ahead.
Step 4. We are halfway through the task. Now, carefully fold beaten egg whites into mascarpone cream.	Next Step.
Step 5. Combine espresso and remaining Amaretto in a shallow dish. Dip ladyfingers in espresso mixture and then layer soaked ladyfingers in bottom of the serving dish.	What are ladyfingers?
Glad you ask! Ladyfingers are low-density, dry, egg-based, sweet sponge cake biscuits roughly shaped like large fingers.	Next Step.
Step 6. Cover with a layer of mascarpone cream and top with a fine layer of grated chocolate.	I’m done.
Step 7. Repeat layering process until all ingredients are used up. Finish up with a layer of mascarpone cream and dust with unsweetened cocoa powder.	Next.
Step 8. This is the last step of this recipe. Refrigerate for at least 3 hours before serving.	Next.
Excellent work, we just finished Classic tiramisu together! Hope you had fun with me today!	Goodbye!

B Conversational Steps Traits Description

The following traits were classified in a 1 to 5 Likert scale considering their importance in a task-guiding setting:

- **Step Complexity** - Was the complexity of the step an important factor to segment?

- **Step Clarity** - Was the clarity of a step important, i.e. information was clear, organized and well delimited in each step?
- **Step Length and Number of Steps** - Did the length of the steps and the total number of steps influence your decision?
- **Ability to Parallelize Tasks** - Were you concerned that the segmentation should allow the user to parallelize multiple tasks/steps?
- **Naturalness** - Was the naturalness of each step important for your decision? i.e., sounded close to a person explaining the task?

C Model Training and Hyperparameters

We trained every model for 20 epochs and evaluated in the test set the model with the best performance in the validation set in terms of F-Score. We used a batch size of 16, a learning rate of 5^{-5} , and the Adam optimizer (Kingma and Ba, 2015). All models were trained on a single NVIDIA A100 GPU, except for T5-3B (Enc-only), which used 4 NVIDIA A100 GPUs. We also highlight that these models are faster to train compared to the cross-segment model (Lukasik et al., 2020), due to each training sample only predicting one segmentation, instead of all segments as in DTS.

D Model Output Examples

Table 7 shows examples comparing the original web version of a recipe with the output of the T5-3B (Enc-only) DTS model. As we can observe, the model outputs a larger number of breaks complementing our findings in this conversational scenario. The model is also able to keep a notion of the sub-task being performed, for instance in Example-3 Step-4, the model does not segment the sentences into multiple steps.

Example 1 - <i>Soy Garlic Steak</i> (Web)	Model Output
<ol style="list-style-type: none"> 1. In a small bowl, mix vegetable oil, soy sauce, vinegar, ketchup, and crushed garlic. Place flank steak in a large resealable plastic bag. Pour the marinade over steak. Seal, and marinate in the refrigerator at least 3 hours. 2. Preheat grill for high heat. 3. Oil the grill grate. Place steaks on the grill, and discard marinade. Cook for 5 minutes per side, or to desired doneness. 	<ol style="list-style-type: none"> 1. In a small bowl, mix vegetable oil, soy sauce, vinegar, ketchup, and crushed garlic. 2. Place flank steak in a large resealable plastic bag. Pour the marinade over steak. Seal, and marinate in the refrigerator at least 3 hours. 3. Preheat grill for high heat. 4. Oil the grill grate. Place steaks on the grill, and discard marinade. Cook for 5 minutes per side, or to desired doneness.
Example 2 - <i>Blueberry Yogurt Pops</i> (Web)	Model Output
<ol style="list-style-type: none"> 1. Combine all ingredients in blender. Cover; blend on high speed 15 to 20 seconds or until smooth. 2. Pour into 8 frozen pop molds (2.5 ounces to 3 ounces each). Cover, insert craft sticks, and freeze for 2 hours or until completely firm. 3. To serve, dip outsides of molds into warm water to loosen. 	<ol style="list-style-type: none"> 1. Combine all ingredients in blender. Cover; blend on high speed 15 to 20 seconds or until smooth. 2. Pour into 8 frozen pop molds (2.5 ounces to 3 ounces each). 3. Cover, insert craft sticks, and freeze for 2 hours or until completely firm. 4. To serve, dip outsides of molds into warm water to loosen.
Example 3 - <i>Quinoa Salad with Roasted Yams</i> (Web)	Model Output
<ol style="list-style-type: none"> 1. Preheat oven to 350 degrees F (175 degrees C). Line a baking sheet with aluminum foil; add yams. 2. Bake in the preheated oven until yams are tender and wrinkled at the edges, about 20 minutes. Cool to room temperature, about 15 minutes 3. Bring water to a boil in a large saucepan. Add quinoa, stirring once; return to boil. Cook uncovered until water is absorbed, 10 to 12 minutes. Strain, shaking the sieve well to remove all moisture. Transfer to a mixing bowl. 4. Stir cucumbers, yams, parsley, olive oil, onion, lemon juice, red wine vinegar, salt, and pepper into the quinoa. Garnish with endive spears. 	<ol style="list-style-type: none"> 1. Preheat oven to 350 degrees F (175 degrees C). 2. Line a baking sheet with aluminum foil; add yams. 3. Bake in the preheated oven until yams are tender and wrinkled at the edges, about 20 minutes. Cool to room temperature, about 15 minutes 4. Bring water to a boil in a large saucepan. Add quinoa, stirring once; return to boil. Cook uncovered until water is absorbed, 10 to 12 minutes. 5. Strain, shaking the sieve well to remove all moisture. Transfer to a mixing bowl. 6. Stir cucumbers, yams, parsley, olive oil, onion, lemon juice, red wine vinegar, salt, and pepper into the quinoa. Garnish with endive spears.

Table 7: Examples comparing original web recipe and the T5-3B (Enc-only) DTS model outputs.