OPENPI2.0: An Improved Dataset for Entity Tracking in Texts

Li Zhang[1*] **Hainiu Xu**[1] **Abhinav Kommula**[3]
Chris Callison-Burch[1] **Niket Tandon**[2]
[1]University of Pennsylvania [2]Allen Institute for Artificial Intelligence
[3]University of California, Berkeley
{zharry}@upenn.edu {nikett}@allenai.org

Abstract

Much texts describe a changing world (e.g., procedures, stories, newswires), and understanding them requires tracking how entities change. An earlier dataset, OPENPI, provided crowd-sourced annotations of entity state changes in text. However, a major limitation was that those annotations were free-form and did not identify salient changes, hampering model evaluation. To overcome these limitations, we present an improved dataset, OPENPI2.0, where entities and attributes are fully canonicalized and additional entity salience annotations are added. On our fairer evaluation setting, we find that current state-of-the-art language models are far from competent. We also show that using state changes of salient entities as a chain-of-thought prompt, downstream performance is improved on tasks such as question answering and classical planning, outperforming the setting involving all related entities indiscriminately. We offer OPENPI2.0 for the continued development of models that can understand the dynamics of entities in text.[1]

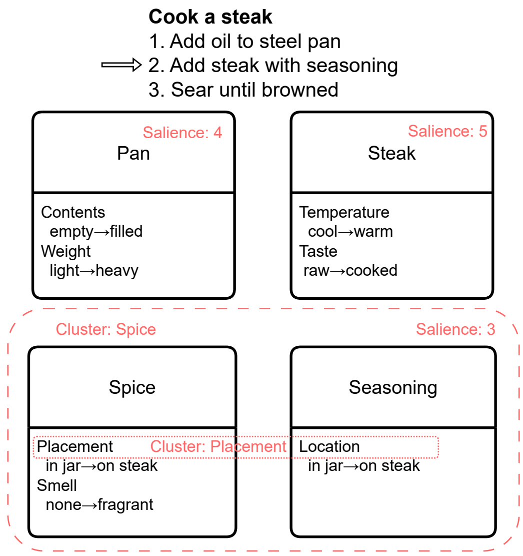

Figure 1: For each step in a procedure, OPENPI annotates the state change of attributes of entities. Our OPENPI2.0 additionally (shown in red boxes and texts) canonicalizes the entities and attributes and includes their salience scores.

1 Introduction

Tracking entity states in procedural texts (Weston et al., 2015; Bosselut et al., 2017; Dalvi et al., 2018) is closely related to many NLP reasoning tasks. To name a few, question answering about events (e.g., *should one use gloves when retrieving the tray from the oven*) often require knowledge of entity states (e.g., *the tray becomes very hot while in the oven; gloves insulate heat*) (Tandon et al., 2019; Spiliopoulou et al., 2022; Zhang et al., 2023a); planning (Wang et al., 2022; Brohan et al., 2023) largely involves actions upon entities resulting in state changes. While most recent work has relied on end-to-end language models (LMs) (Huang et al., 2022), recent work has shown that explicit modeling entity states benefits LMs in such tasks (Zhang et al., 2023a). Procedural entity tracking is challenging in itself, requiring much understanding of an implicit environment as well as external knowledge of events and entities.

We propose the OPENPI2.0 dataset which builds on OPENPI (Open Procedural Inference) (Tandon et al., 2020), a large-scale dataset for tracking entity states in procedural texts. OPENPI contains annotations of entities, attributes, and state changes for each step (e.g., after the step "set the pan in a heated oven", the *pan*'s *temperature* was *cool* before and *hot* afterwards). OPENPI2.0 features two critical improvements (see Figure 1 for a demonstration of key features of OPENPI and OPENPI2.0):

1. **Canonicalization**. Originally, different mentions of the same entity or attribute render evaluation difficult. Here, we prompt LMs to effec-

[*]Work done as an intern at AI2.
[1]Our resources can be found at https://github.com/allenai/openpi-dataset/tree/main/v2.0.

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics Volume 1: Long Papers, pages 166–178
March 17-22, 2024

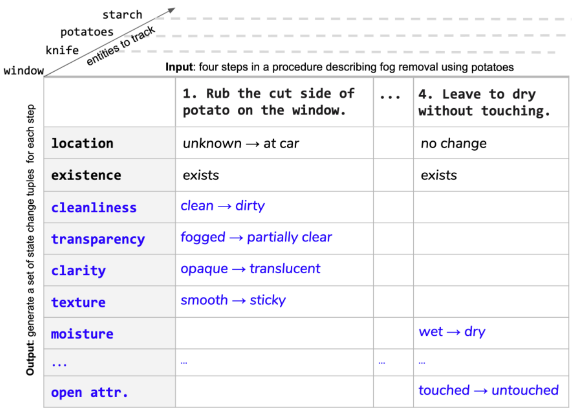

Figure 2: An example from the original OPENPI dataset (Tandon et al., 2020).

tively cluster the entities and attributes.

2. **Entity Salience**. Originally, a large amount of entities that undergo changes are listed in parallel. Here, we provide both human and model-predicted annotations of their salience.

Regarding canonicalization, clustering different mentions (e.g., coffee maker, espresso machine) of the same entity allows for fairer evaluation. Moreover, as our task of predicting entities, attributes, and states is a generation task with imperfect and incomplete ground-truth references, we show that expanding each entity or attribute cluster with possible paraphrases (thus providing more references) is effective for reducing the false-negative rate. We then comprehensively report various state-of-the-art LMs' performance of entity tracking on OPENPI2.0.

Regarding entity salience, we provide both manually annotated and automatically predicted labels. We evaluate them based on correlation with ground-truth data, and show that LMs can reliably predict entity salience with a close-to-human performance. We argue that salient entities act as a means of compression of the most critical information in procedural texts, similar to saliency maps in computer vision (Simonyan et al., 2013). We proceed to qualitatively and quantitatively show that salient entities, as chain-of-though of LM prompting, benefit downstream tasks such as question answering and classical planning, while reducing cost by excluding less important entities in the prompt.

OPENPI2.0 have following advantages:

1. The canonicalization of entities and attributes (§3) that facilitates evaluation (§3.2);
2. The salience of entities (§4) that improves performance on downstream tasks (§4.3).

2 The Original OPENPI Dataset

Our work OPENPI2.0 builds upon the OPENPI dataset (Tandon et al., 2020) that tracks entity state changes in procedural texts with an open vocabulary. The procedures are extracted from wikiHow, a web resource containing instructions of everyday tasks. As exemplified in Figure 2, the input is a procedure which includes a goal (e.g., "remove fog using a potato") and a sequence of ordered steps (e.g., "rub the cut side of potato on the window"). For each step, the output is an array of 4-tuples describing an entity state change. Each 4-tuple contains an entity, an attribute, a state before the step, and a state after the step (e.g., *the window*'s *texture* was *smooth* before and *sticky* after). The task is thus equivalent to predicting an entity state matrix given a procedure, where the axes are step, entity, and attribute, while the value is the before and after states. The data is annotated via crowdsourcing and manually validated.

However, OPENPI lacks canonicalization of entities and the differentiation of salient entities. In our work of OPENPI2.0, we will address both issues using state-of-the-art models.

3 Canonicalization

In the original OPENPI dataset, the entities and attributes that undergo change were written by crowd workers. Consequently, the dataset contains different ways of expressing the same entity (e.g., *coffee maker*, *coffee machine*, *espresso machine* in a coffee-making procedure) or attribute (e.g., *texture*, *smoothness*, *sheen* of a paint). Canonicalization by clustering the entities and procedures is thus important for two reasons: 1) it facilitates evaluation, especially in a generation setting, where a model might be wrongly penalized for predicting the paraphrase of some correct entity or attribute; 2) it facilitates further annotation of features such as salience (§4) of the entities and attributes. Here, we describe efforts to canonicalize the entities and attributes in the evaluation set of OPENPI.

3.1 Clustering Entities and Attributes

While canonicalization seems straightforward, it is non-trivial in OPENPI2.0 because clustering is highly context-dependent. For example, the entity *torso* and *paper chunk* usually have nothing to do with each other, but in fact refer to the same thing in a procedure of "making a paper bird."

Role	Content
User	I am trying to make coffee. First, I put some coffee beans and tap water into the corresponding compartment of the espresso machine. Then, I select the desired type of coffee to make produced. Then I put a mug under the espresso machine and press start. Do you get it?
Assistant	Yes.
User	We have the following objects: *water*, *coffee maker*, *coffee machine*, *mug*, *espresso machine*. Group those that refer to the same thing. You must include all the provided entities. Do not add any entity that is not provided in the list.
Assistant	<start of generation> The grouped objects are: - ['water'] - ['coffee maker', 'coffee machine', 'espresso machine'] - ['mug']

Table 1: Our chosen prompt for entity and attribute clustering.

	Entity	Attribute
Cluster Recall	.425	.881
Cluster Precision	.593	.906
Cluster F1	.495	.893

Table 2: Evaluation of entity and attribute clustering.

Clustering Due to the contextual nature of the task, we prompt one of the state-of-the-art LMs `gpt-3.5-turbo` (a.k.a. ChatGPT)[2] as shown in Table 1. We use 3-shot prompting, meaning that the complete prompt includes three handwritten examples and the prompt header of the example to be inferred, only containing the "User" role. The temperature is as default (0.7) and so are other hyperparameters. We aggregate output from five runs of ChatGPT as the final entity cluster and three runs for attribute cluster, as doing so is found to be empirically superior than a single-pass generation.[3]

To see if our model can cluster entities and attributes effectively, we evaluate the results using cluster-level precision, recall, and F1 scores with exact match against a set of manually-labeled clusters from 20 procedures in the development set.

We see that ChatGPT scores better in clustering attributes compared to entities. Error analysis shows that two factors contribute to this performance discrepancy. First, most attributes describe the physical properties of an entity. Therefore, attribute clusters are less context-dependent compared to entity clusters. Second, many attributes are shared amongst entities. For instance, out of 1,145 attribute annotations in the development set, 204 of them are *"location"*.

Cluster expansion Though the existing entities and attributes are now clustered in OpenPI2.0, there may still be other paraphrases that a model might rightfully predict and wrongfully penalized for. Thus, we again prompt ChatGPT to expand the clusters by generating paraphrases given a cluster of entities or attributes (prompt omitted).

To evaluate the quality of entities and attributes generated from the expansion, we manually rate 20 procedures and find that 83.3% of the generated, paraphrased entities and 59.4% attributes are correct. This is largely because entity names are oftentimes self-explanatory and less context-dependent whereas the attribute names and their meanings are highly dependant on the context.

3.2 Utility: Evaluation of Entity Tracking

Just as the original evaluation set of OpenPI, OpenPI2.0 is meant to benchmark models on entity tracking – given a step in a procedure, predicting the state changes that entities and their attributes undergo. With the entities and attributes in OpenPI2.0 now fully canonicalized, evaluation can be done more fairly. To start with, we follow Tandon et al. (2020) and have models predict one **complete** sentence: "*attribute* of *entity* is *pre-state* before and *post-state* afterwards", which is then compared to such sentences in the ground-truth data (Table 4). We further make the evaluation more fine-grained by formulating two sub-tasks: i. predicting **schemata**, namely the entities and their corresponding attributes given a step (e.g., given "turn on the oven", the *temperature* of the *rack* undergo state changes), and ii. predicting the change of **states** given a step, an entity and an attribute (e.g., given the previous information, the state change is from *cool* to *hot*). This evaluation

[2] platform.openai.com/docs/models/gpt-3-5

[3] With results from multiple runs, entity clusters are greedily selected based on their number of occurrences. For instance, if `(pan, cookware, container)` occurred four times whereas `(pan, pot)` just once, then the former will be added to the final cluster.

	schemata (global)				schemata (local)				states	
	F1	F1 + exp	BS	BS + exp	F1	F1 + exp	BS	BS + exp	acc.	BS
gpt-3.5-turbo	.151	.249	.843	.869	.025	.039	.798	.804	.074	.600
text-davinci-003	.362	.450	.891	.920	.130	.155	.798	.810	.225	.682
LLaMA 65B	.129	.174	.799	.820	.045	.060	.801	.800	.102	.577

Table 3: Exact match F1 or accuracy and BERTScore on the schemata and states prediction sub-tasks, with and without cluster expansion. The schemata sub-task is evaluated both globally (per-procedure) and locally (per-step).

	complete			
	F1	F1+exp	BS	BS+exp
gpt-3.5-turbo	.016	.016	.772	.790
text-davinci-003	.034	.034	.807	.821
LLaMA 65B	.117	.117	.429	.440

Table 4: Exact match F1 and BERTScore of complete sentences including an entity, an attribute, a pre-state, and a post-state, following the original OPENPI paper. Canonicalization and expansion lead to little help for exact match as it is only done on entity and attribute clusters, while the state names can still be expressed in many ways, causing false negatives.

	Correct	No change	Nonsense	Missing
003	585 (82.3%)	106 (15.0%)	14 (2.0%)	383 (20.3%)
3.5	303 (59.4%)	173 (33.9%)	34 (6.7%)	218 (42.7%)

Table 5: Error analysis on the schemata prediction task of `text-davinci-003` and `gpt-3.5-turbo`.

of first predicting a skeleton tensor of entities and attributes is highly practical, with a notable advantage over previous work (§6) in closed-domain entity tracking, where states are predicted using *given* entities and attributes.

On the development set, we run three state-of-the-art LMs: `gpt-3.5-turbo`, `text-davinci-003`[4] (Brown et al., 2020), and the open-source LLaMA 65B (Touvron et al., 2023). For each model, we start by separately tackling each of the two sub-tasks[5]; namely, a model first predicts attributes of entities (schemata) given a step, and then predicts a pre-state and a post state (states) given the gold entity-attribute pair. All experiments are via 1-shot prompting. See details on prompt formulation in Appendix B.

For all settings, we consider both exact match (F1 for schemata and complete sentence prediction and accuracy for states prediction) and BERTScore (Zhang et al., 2020d) based on `deberta-xlarge-mnli` (He et al., 2021).

For the schemata prediction sub-task (Table 3), the atomic unit to be evaluated is an entity-attribute pair. We consider both a *global* evaluation, where predictions are made per-procedure (e.g., what attributes of entities undergo state changes in the procedure), and a *local* evaluation, where predictions are made per-step. This categorization will reappear in §4.2. Schemata prediction is naturally influenced by our entity and attribute clusters. Hence, for exact match we report F1 scores based on exact matches where any entity-attribute prediction that falls under an cluster, obtained by taking a Cartesian product of an entity cluster and an attribute cluster, is considered a true positive. For BERTScore, we calculate the maximum score of a prediction against all entity-attribute strings within all ground-truth clusters. Then, we report the mean score among all predictions as a macro average.

The states prediction sub-task (Table 3) is much more straightforward as the entity-attribute pairs are provided and a model only needs to predict a pre-state and a post-state for each. Thus, we simply report the exact match accuracy and BERTScore for each state.

3.3 Discussion and Error Analysis

We observe that the predicting attributes of entities that undergo state changes is a highly challenging task even for state-of-the-art LMs. Although evidently, expansion of clusters improves performance (fairly, as we have shown that the generated paraphrases are mostly correct), false-negatives that result in underestimation of models cannot be eliminated entirely. One interesting observation is that `text-davinci-003` greatly outperforms the supposedly more superior `gpt-3.5-turbo`. To gain even more insights into models' behavior, we analyze the model output for the schemata prediction sub-task. For each step, we annotate each entity-attribute prediction based on three labels:

- Correct, where the entity-attribute indeed go

[4] `platform.openai.com/docs/models/gpt-3-5`

[5] To avoid error propagation, for states prediction, the ground-truth entities and attributes are provided.

Role	Content
User	Here are some instructions on making coffee. - Buy fresh coffee beans. - Grind the coffee beans. - ... Now, I will provide you with a series of objects, and you will assign scores on a scale of 1-5 to them based on their importance in the instruction. Your answer should strictly be a numerical score followed by a one-sentence explanation.
Assistant	Sure, I can help you with that. Please provide the objects.
User	Coffee bean
Assistant	<generation> 5 - the coffee beans are the most important ingredient in making coffee.

Table 6: Our chosen prompt for predicting global or procedure-wide entity salience. For local salience, the wording is similar with only one step provided.

through some changes;

- Incorrect, because the entity-attribute actually does not go through any changes;
- Incorrect, because the entity-attribute is nonsensical.

In addition, we add any entity-attribute pairs that should have been predicted as going through some change, to measure models' recall. We randomly sample 20 procedures to perform this error analysis and the results are shown in Table 5.

Regarding precision, we find that while the majority of the predicted entities are correct, many of the predicted associated attributes are generic ones that do not undergo any change either locally or globally. For example, for the step "Purchase a blackboard eraser", the attributes predicted by `text-davinci-003` for the entity *eraser* are *location* (correct), *cleanness* (no change locally), *shape*, and *size* (no change globally). The issue is much more pronounced with `gpt-3.5-turbo`, with predictions such as *location* of *seller*, *name* of *brand*, etc, despite that the prompt clearly explains the desired output with an example. We attribute such performance discrepancy to `gpt-3.5-turbo`'s decreased ability to follow examples and its inability to understand nuanced instructions. Regarding recall, both models fail to predict many attributes that the human annotator deems changing. Upon qualitative inspection, most of these missing attributes are no less salient than the predicted ones.

We leave to future work the resolution of these issues, which can be mitigated by re-prompting the models by validating if the predicted attributes indeed undergo changes, or simply have them predict the state changes altogether in the first place.

	Annotations Human (A2)	Predictions LM
Global	.759	.719
Local	.578	.400

Table 7: Pearson' r between model prediction and human annotations (A1) of entity salience.

4 Salience

The original OpenPI is annotated with many parallel entities in each procedure. Often, they vary greatly by importance in accomplishing the task. For example, in a procedure of "cooking a steak", entities *fish*, *oven*, *gloves*, and *spice rack* might all be involved, while some are more indispensable than the rest. Intuitively, the knowledge of entity salience helps models focus on what matters in downstream tasks (§4.3). In OpenPI2.0, we define two types of entity salience: the global salience refers to the importance of an entity in accomplishing the goal of the procedure, whereas the local salience refers to that in a step.

4.1 Annotations

Human Labeling To first procure ground-truth salience labels, two experts (referred to as A1 and A2) annotated entity salience in the first 20 procedures in the development set as the gold standard of entity salience. We devise and follow these annotation criteria in a Likert scale:

5: without or without mentioning this entity, the procedure or step cannot be done at all (e.g., *lemon* in "Wash faucet with lemon")

4: without this entity, another entity of the same type can be used as a replacement, perhaps with worse outcome or more efforts (e.g., pan in "Sear a salmon" - can also use *grill*)

3: without this entity, the procedure or step can

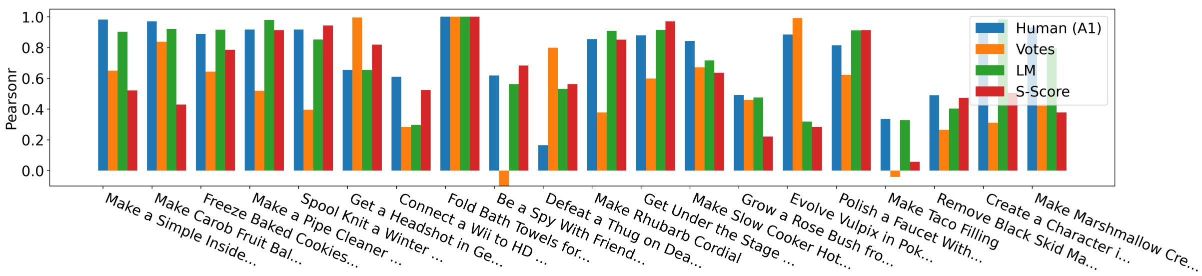

Figure 3: Per-procedure correlation of global entity salience between each set of annotations and the ground-truth human annotations.

be done in principal, though with slightly worse outcome or more efforts (e.g., *glove* in "Cut off tough branches of a bonsai plant")

2: without this entity, the procedure or step can be done, though with negligibly worse outcome or more efforts (e.g., *vacuum cleaner* in "Drill holes in the wall")

1: the entity appears in the procedure or step rather gratuitously, and the lack thereof makes no difference

0: the entity is irrelevant to the procedure or step

Subjectivity is inevitable even though we strive to minimize subjectivity using this fine-grained scale to capture nuanced situations (e.g., an entity that frequently appears that can be easily replaced versus one that appears only once but is irreplaceable). In later sections, we will see how this scale leads to reasonable inter-annotator agreement and favorable performance on downstream tasks.

LM Prediction We prompt `gpt-3.5-turbo`, as before, to automatically predict salience. Table 6 shows an example prompt for predicting global salience. As before, we use the default hyperparameters with a temperature of 0.7. We parse the result by extracting the first digit from the generation as the score, and default to 1 whenever impossible.

4.2 Evaluation

To first holistically evaluate the modelling of salience, we report pairwise Pearson's correlation coefficients between each set of labels above and the annotations of human A1. In Table 7, we report a "macro correlation", namely the mean of correlation of salience scores in each procedure.[6] First, the correlation between the two annotators is high but imperfect, implying subjectivity in the annotation of entity salience. In comparison, the LM predictions come close with especially impressive predictions for global salience.

To understand when and how entity salience can be subjective among humans, in Figure 3 we show salience correlation for the first 20 procedures. Some with low inter-human correlation such as "Defeat a Thug" expose a document-frequency problem: one human labels the entity *you (actor)* with a salience of 5, believing that without the *actor* the procedure cannot be undertaken at all, while the other labels 1, believing that the *actor* occurs in every procedure and thus the mention of it makes no difference. Other procedures such as "Make Taco Filling" expose a cluster-individual problem: all individual ingredients are grouped under one cluster, for which annotators are divergent on the salience. Both issues give rise to subjectivity and can be mitigated by better defining entity salience in a potential future annotation process.

4.3 Utility: Downstream Applications

We argue that entity salience is an effective way to compress information expressed by procedural texts. In other words, states of the most salient entities are sufficient for downstream tasks where entity tracking can be applied to. We provide both qualitative and quantitative evidence on two datasets both in the domain of procedural texts.

4.3.1 Question Answering

CREPE (Zhang et al., 2023a) is a dataset for procedural question answering and causal reasoning. Given a procedure (e.g., steps of "baking a cake"), a model predicts the change of likelihood of some event occurrence (e.g., "there is a pleasant smell coming from the oven") after the execution of each step (e.g., "set the tray in the oven for 45 minutes").

First, we show a qualitative example in Table 8 in the format of CREPE. Clearly, the model's third response is wrong, and the user's first utterance im-

[6]To avoid NaN due to constant input array, a 0 is appended to each array as smoothing.

Role	Content
User	A person walks into a dark room in a **freezing windy night**. In the very beginning before anything is done, how likely is it that she can see what's in the room? Answer on a scale of 1-5, with a 1-sentence explanation.
Assistant	1 - it is highly unlikely for the person to see anything in a dark room without any source of light.
User	Next, she lights up a torch and hangs it near a window. At this point, how likely (same question)...
Assistant	5 - hanging the torch near a window maximizes the distribution of light in the room, greatly enhancing the visibility and increasing the likelihood of seeing what's inside.
[1]User	Next, she **opens the window**. At this point, how likely (same question)...
[1]Assistant	**5** - opening the window allows **natural light** to enter the room, combining with the torchlight to provide ample illumination, greatly increasing the likelihood of seeing what's in the room.
[2]User	Next, she opens the window. The most salient entities for this step are the window, the wind, and the flames. How does each of these entities change?
[2]Assistant	When she opens the window, the following changes occur: Window: The **window is now open**, allowing for air and potentially some outside light to enter the room. Wind: The wind becomes more noticeable and **enters the room through the open window**, causing a draft and potentially affecting the flames of the torch. Flames: The flames of the torch may flicker or change direction due to the entry of wind through the open window, **altering their shape and intensity**.
[2]User	With such information, at this point, how likely is it that she can see what's in the room?
[2]Assistant	**3** - opening the window allows some outside light to enter, the presence of wind and potential **disruption to the flames** may still impede visibility to some extent.

Table 8: `gpt-3.5-turbo`'s performance on a CREPE-like example. The interactions with and without using entity salience are both shown. Critical information is illustrated in bold.

	dev	test	num. ents per step
ChatGPT	.348	.362	-
+all ents	.392	.369	2.9
+ents sal>=5	**.402**	**.370**	1.8

Table 9: Model performance on CREPE, with and without including state changes of salient entities.

plies that the *torch* would have been *extinguished* by the cold wind or at least flickering, making it harder to see what's in the room. In contrast, the model falsely asserts that there would be *natural light*, which is impossible given the procedure occurs at night. However, if we use the same prompt as §3.2 and Table 6 to first get access to the most locally salient entities, have the model predict their state changes, and use the such information as a chain-of-thought, the model is able to answer correctly. Specifically, the model now recognizes that the *window* is *open*, causing the wind to enter the room, in turn causing the *flames* to *flicker*. With this information equipped, the final predicted answer is now correct as the likelihood to see clearly in the room has decreased. Note that the step of opening the window also involves many other less salient entities, such as *person*, *hands*, *windowsill*, *smell*, etc., which are filtered out by predicted salience.

Following the qualitative study, we then quantitatively evaluate a `gpt-3.5-turbo` model on CREPE in a zero-shot manner. As the labels in CREPE are ternary (an event is more, less, or equally likely to happen after a step), we directly prompt the model to provide one of these labels instead of giving a likelihood score as in Table 8, while keeping the rest of the designs the same. We compare three settings, in which for each step given an event, the model:

1. makes a prediction directly;
2. is provided with all entity state changes (based on §3.2), before making a prediction;
3. is provided with state changes of entities with a predicted salience score (based on §4.1) no less than a threshold, before making a prediction.

Table 9 clearly shows that provided entity state changes as chain-of-though in the prompt improves the performance. Moreover, excluding information of less salient entities does not decrease, and even in some cases improve performance. By only keeping the entities that have a predicted salience score of 5, which constitute about half of all entities, the prompt becomes shorter and thus reducing cost by about 14% for the entire experiment, while achieving slightly higher performance.

4.3.2 Formal Planning based on Texts

A strand of work has focused on using LMs to plan in a structured and symbolic manner (Liu et al., 2023; Xie et al., 2023). Specifically, the task is to

User

Provide a PDDL domain file representing a procedure "Survive in rainy woods":
1. Find natural shelter using the environment
2. Set up containers in the open
3. Stay dry during a rain
4. Stay hydrated
After that, also provide a problem file of "collect water to drink".

Assistant

Here's a domain file:
(define (domain survive-in-rainy-woods)
  requirements, types, predicates are omitted
  (:action find-shelter
    :parameters (?shelter - shelter ?loc - location)
    :precondition (at ?shelter ?loc)
    :effect (and (sheltered ?shelter)))
  (:action set-up-container
    :parameters (?container - container ?loc - location)
    :precondition (and (at ?container ?loc) (sheltered ?container))
    :effect (and (contains **?container ?water**)))
  (:action stay-dry-during-rain
    :parameters (?obj - object ?loc - location)
    :precondition (and (at ?obj ?loc) (raining ?loc))
    :effect (and (not (raining ?loc)) (dry ?obj)))
  (:action stay-hydrated
    :parameters (?obj - object ?container - container)
    :precondition (and (at ?obj ?container) (contains **?container ?water**))
    :effect (and (not (contains ?container ?water)) (**hydrated ?obj**))))
Here's a problem file:
(define (problem collect-water)
  objects are omitted
  (:init
    (at water-bottle woods)
    (at shelter woods)
    (at player woods)
    (raining woods)
    (collecting-water woods))
  (:goal (**hydrated player**)))

Table 10: gpt-3.5-turbo's performance on a wikiHow-to-PDDL-like example. The interactions without using entity salience are both shown. Critical information is illustrated in bold.

convert a a textual description of a procedure to a planning domain definition (PDDL), which can be deterministically solved to find a plan. An instance of PDDL consists of a domain file, which models the pre-conditions and effects of permitted actions, and a problem file, which models the entity states initially and eventually. Here, we show a qualitative example of how identification of salient entities can improve the quality of a generated domain file that models actions.

Table 10 shows an example where an LM provides a domain file based on the procedure "Survive in the woods" and a problem file based on the task "collect water to drink". Clearly, the goal configuration `(hydrated player)` can only be realized via the action `stay-hydrated`, whose pre-condition `(contains ?container ?water)` can only be satisfied in the second step. However, the LM fails to recognize that the third step "Stay dry during a rain" has the byproduct effect of filling up the containers set up in the second step. In this case, we might simply first provide the LM the salient entities for each step, and the LM successfully fixes the third action `stay-dry-during-rain` so that it has the effect of containing containing water. Therefore, the problem file can now be solved reasonably with a sequence of all four actions. We leave to future work a larger-scale experiment of the application of salient entity states to planning.

5 Resulting Dataset: OpenPI2.0

By adding canonicalization of entities and attributes as well as salience of entities to the evaluation set of the OpenPI dataset, we now fully present OpenPI2.0. As the procedures and entity state annotations have not changed, OpenPI2.0 still has 55 procedures with 5.0 steps on average. These procedures are collected from wikiHow and their topics are everyday activities. OpenPI2.0 also inherits the original entity-attribute-state changes annotated by crowd workers. After canonicalization, there are 356 canon entities each with 7.6 unique mentions and 5.5 expanded mentions on average, 3240 canon attributes, each with 3.0 unique mentions and 3.3 expanded mentions on average, and 1193 before-after states in the development set. The quality of clustering and expansion and be evidenced in §3.1. Regarding salience labels (on a scale of 1 to 5), the global salience of entities has a mean of 3.5 and standard deviation of 1.4; the local salience of entities has a mean of 3.4 and standard deviation of 1.5.

6 Related Work

Entity State Tracking Prior work on entity state tracking spans various disciplines of AI. For instance, object tracking, a sub-task of entity state tracking, has led to much work in both robotics (Wang et al., 2007) and computer vision (Comaniciu et al., 2003). In NLP, early efforts focus on synthetic, closed-domain data (Weston et al., 2015; Long et al., 2016) and more recent ones shift attention to real-world procedures (Bosselut et al., 2017; Dalvi et al., 2018; Gupta and Durrett, 2019;

Du et al., 2019; Mysore et al., 2019) with a closed set of entities and attributes. The only open-ended dataset to our knowledge is still OPENPI (Tandon et al., 2020) which we build on.

Entity Salience A small body of work on entity salience has focused on annotating entity salience in news articles and web pages for better information retrieval, recommendation, and linking Gamon et al. (2013); Dunietz and Gillick (2014); Dojchinovski et al. (2016); Trani et al. (2018); Wu et al. (2020). In contrast, we focus on entities in procedural texts, situating our work in script learning, robotic execution, automatic planning and reasoning, etc. Due to this mismatch of purpose, the definition, annotation process, and downstream applications of our entity salience and theirs are all fundamentally different.

Procedures and Scripts Script learning (Schank, 1977) is an umbrella discipline that focuses on groups of human actions under certain scenarios. Regarding domain, procedural texts are an attractive data source to reason about entities which undergo frequent changes. There has been steady efforts in computer vision (Miech et al., 2019), robotics (Brohan et al., 2023), and language (Mujtaba and Mahapatra, 2019; Zhang, 2022). In NLP specifically, work on procedures includes extracting them from instructional texts (Paris et al., 2002; Delpech and Saint-Dizier, 2008; Zhang et al., 2012), reasoning about relations among events (Takechi et al., 2003; Tandon et al., 2019; Rajagopal et al., 2020; Zhang et al., 2020c, 2023b), knowledge-base construction (Jung et al., 2010; Chu et al., 2017; Park and Motahari Nezhad, 2018; Zhou et al., 2022), or applying them to downstream applications (Yang et al., 2021b,a; Zhang et al., 2020a; Lyu et al., 2021; Dalvi et al., 2019; Zhang et al., 2020b; Chen et al., 2020). As discussed in many of these cited works, knowledge acquired from learning scripts and procedures has been known to benefit robotics and planning.

7 Conclusion

We propose OPENPI2.0, an improved dataset on open-domain entity tracking in procedural texts. OPENPI2.0 features canonicalization of entities and attributes, based on which we perform a comprehensive benchmarking evaluation of state-of-the-art LMs. OPENPI2.0 also provides human annotation, model prediction, and analyses of entity salience, using which we show qualitative examples on its effective on various downstream tasks.

Limitations

OPENPI2.0, just like its predecessor OPENPI, includes procedures from wikiHow which may result in homogeneous domains, writing styles, and potentially though unlikely biased, erroneous or unsafe information. Regarding canonicalization, due to the limitation of models and the imperfect human annotations in OPENPI, there still exists false negatives while evaluating with metrics based on exact-match. Regarding entity salience, the definition of "how indispensable an entity is in executing the procedure" is motivated empirically downstream tasks and may benefit from refinement or theoretical support. The evaluation could be more trustworthy given more annotators and more procedures to be annotated. The chosen downstream tasks in this work might not be representative of all use cases of entity tracking.

Acknowledgements

We thank Yash Kumar Lal for providing insights into model performance, and Peter Clark for help with presentation of this paper. This work is supported in part by the DARPA KAIROS Program (contract FA8750-19-2-1004), AFRL (contract FA8750-23-C-0507), the Office of the Director of National Intelligence (ODNI) via the IARPA HIATUS Program (contract 2022-22072200005), the NSF (Award 1928631), and gifts from Roblox and Salesforce. Approved for Public Release, Distribution Unlimited. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, ODNI, IARPA, NSF, AFRL, the U.S. Government, or of Roblox or Salesforce. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

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

Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. 2023.

Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, pages 287–318. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Cuong Xuan Chu, Niket Tandon, and Gerhard Weikum. 2017. Distilling task knowledge from how-to communities. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 805–814. ACM.

Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. 2003. Kernel-based object tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 25(5):564–577.

Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1595–1604, New Orleans, Louisiana. Association for Computational Linguistics.

Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, and Peter Clark. 2019. Everything happens for a reason: Discovering the purpose of actions in procedural text. 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 4496–4505, Hong Kong, China. Association for Computational Linguistics.

Estelle Delpech and Patrick Saint-Dizier. 2008. Investigating the structure of procedural texts for answering how-to questions. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Milan Dojchinovski, Dinesh Reddy, Tomáš Kliegr, Tomáš Vitvar, and Harald Sack. 2016. Crowdsourced corpus with entity salience annotations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3307–3311, Portorož, Slovenia. European Language Resources Association (ELRA).

Xinya Du, Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, Peter Clark, and Claire Cardie. 2019. Be consistent! improving procedural text comprehension using label consistency. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2347–2356, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dunietz and Daniel Gillick. 2014. A new entity salience task with millions of training examples. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 205–209, Gothenburg, Sweden. Association for Computational Linguistics.

Michael Gamon, Tae Yano, Xinying Song, Johnson Apacible, and Patrick Pantel. 2013. Identifying salient entities in web pages. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2375–2380.

Aditya Gupta and Greg Durrett. 2019. Tracking discrete and continuous entity state for process understanding. In *Proceedings of the Third Workshop on Structured Prediction for NLP*, pages 7–12, Minneapolis, Minnesota. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR.

Yuchul Jung, Jihee Ryu, Kyung-min Kim, and Sung-Hyon Myaeng. 2010. Automatic construction of a large-scale situation ontology by mining how-to instructions from the web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(2-3):110–124.

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. 2023. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*.

Reginald Long, Panupong Pasupat, and Percy Liang. 2016. Simpler context-dependent logical forms via model projections. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1456–1465, Berlin, Germany. Association for Computational Linguistics.

Qing Lyu, Li Zhang, and Chris Callison-Burch. 2021. Goal-oriented script construction. In *Proceedings of* *the 14th International Conference on Natural Language Generation*, pages 184–200, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640.

Dena Mujtaba and Nihar Mahapatra. 2019. Recent trends in natural language understanding for procedural knowledge. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 420–424.

Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 56–64, Florence, Italy. Association for Computational Linguistics.

Cécile Paris, Keith Vander Linden, and Shijian Lu. 2002. Automated knowledge acquisition for instructional text generation. In *Proceedings of the 20th Annual International Conference on Computer Documentation*, SIGDOC '02, page 142–151, New York, NY, USA. Association for Computing Machinery.

Hogun Park and Hamid Reza Motahari Nezhad. 2018. Learning procedures from text: Codifying how-to procedures in deep neural networks. In *Companion Proceedings of the The Web Conference 2018*, pages 351–358.

Dheeraj Rajagopal, Niket Tandon, Peter Clark, Bhavana Dalvi, and Eduard Hovy. 2020. What-if I ask you to explain: Explaining the effects of perturbations in procedural text. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3345–3355, Online. Association for Computational Linguistics.

Roger C. Schank. 1977. *Scripts, plans, goals, and understanding : an inquiry into human knowledge structures /*. L. Erlbaum Associates ;, Hillsdale, N.J. :.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Evangelia Spiliopoulou, Artidoro Pagnoni, Yonatan Bisk, and Eduard Hovy. 2022. EvEntS ReaLM: Event reasoning of entity states via language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1982–1997, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mineki Takechi, Takenobu Tokunaga, Yuji Matsumoto, and Hozumi Tanaka. 2003. Feature selection in categorizing procedural expressions. In *Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages*, pages 49–56, Sapporo, Japan. Association for Computational Linguistics.

Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. WIQA: A dataset for "what if..." reasoning over procedural text. 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 6076–6085, Hong Kong, China. Association for Computational Linguistics.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. A dataset for tracking entities in open domain procedural text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417, Online. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Salvatore Trani, Claudio Lucchese, Raffaele Perego, David E Losada, Diego Ceccarelli, and Salvatore Orlando. 2018. Sel: A unified algorithm for salient entity linking. *Computational Intelligence*, 34(1):2–29.

Chieh-Chih Wang, Charles Thorpe, Sebastian Thrun, Martial Hebert, and Hugh Durrant-Whyte. 2007. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26(9):889–916.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. Scienceworld: Is your agent smarter than a 5th grader? *arXiv preprint arXiv:2203.07540*.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Chuan Wu, Evangelos Kanoulas, Maarten de Rijke, and Wei Lu. 2020. WN-salience: A corpus of news articles with entity salience annotations. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2095–2102, Marseille, France. European Language Resources Association.

Yaqi Xie, Chen Yu, Tongyao Zhu, Jinbin Bai, Ze Gong, and Harold Soh. 2023. Translating natural language to planning goals with large-language models. *arXiv preprint arXiv:2302.05128*.

Yue Yang, Joongwon Kim, Artemis Panagopoulou, Mark Yatskar, and Chris Callison-Burch. 2021a. Induce, edit, retrieve: Language grounded multimodal schema for instructional video retrieval. *ArXiv preprint*, abs/2111.09276.

Yue Yang, Artemis Panagopoulou, Qing Lyu, Li Zhang, Mark Yatskar, and Chris Callison-Burch. 2021b. Visual goal-step inference using wikiHow. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2167–2179, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hongming Zhang, Muhao Chen, Haoyu Wang, Yangqiu Song, and Dan Roth. 2020a. Analogous process structure induction for sub-event sequence prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1541–1550, Online. Association for Computational Linguistics.

Li Zhang. 2022. Reasoning about procedures with natural language processing: A tutorial.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020b. Intent detection with WikiHow. 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*, pages 328–333, Suzhou, China. Association for Computational Linguistics.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020c. Reasoning about goals, steps, and temporal ordering with WikiHow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4630–4639, Online. Association for Computational Linguistics.

Li Zhang, Hainiu Xu, Yue Yang, Shuyan Zhou, Weiqiu You, Manni Arora, and Chris Callison-Burch. 2023a. Causal reasoning of entities and events in procedural texts. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 415–431, Dubrovnik, Croatia. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020d. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Tianyi Zhang, Isaac Tham, Zhaoyi Hou, Jiaxuan Ren, Liyang Zhou, Hainiu Xu, Li Zhang, Lara J. Martin, Rotem Dror, Sha Li, Heng Ji, Martha Palmer, Susan Brown, Reece Suchocki, and Chris Callison-Burch. 2023b. Human-in-the-loop schema induction.

Ziqi Zhang, Philip Webster, Victoria Uren, Andrea Varga, and Fabio Ciravegna. 2012. Automatically extracting procedural knowledge from instructional texts using natural language processing. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 520–527, Istanbul, Turkey. European Language Resources Association (ELRA).

Shuyan Zhou, Li Zhang, Yue Yang, Qing Lyu, Pengcheng Yin, Chris Callison-Burch, and Graham Neubig. 2022. Show me more details: Discovering hierarchies of procedures from semi-structured web data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2998–3012, Dublin, Ireland. Association for Computational Linguistics.

A Licensing

Our proposed OpenPI2.0 dataset, along with the OpenPI and CREPE datasets used in this work, are under MIT license. These datasets have been and should be accessed for research purposes. During the creation of OpenPI2.0, we made sure that no PII is included.

B Prompts of Evaluation of Entity Tracking

Example prompts pertaining to §3.2 are shown in Table 11, 12, 13, and 14.

Content
A person's goal is to bake a cake. For each of the steps, list the involved entities and attributes THAT UNDERGO ANY CHANGE. For example, for the step 'heat the oven', rack (temperature) is correct, while oven (color) is wrong. Step: Mix the eggs with flour. Entities and attributes: <generation> eggs (shape), flour (color, location), mixing bowl (content, weight)

Table 11: Our prompt for `text-davinci-003` for the schemata prediction sub-task, followed by 1-shot demonstration.

Role	Content
User	A person's goal is to bake a cake. For each of the steps, you will list entities and attributes THAT UNDERGO ANY CHANGE. For example, for the step 'heat the oven', rack (temperature) is a good answer, while oven(color) is a bad answer. Are you ready?
Assistant	Yes, I'm ready.
User	Step: Mix the eggs with flour.
Assistant	<generation> eggs (shape), flour (color, location), mixing bowl (content, weight)

Table 12: Our prompt for `gpt-3.5-turbo` for the schemata prediction sub-task, followed by 1-shot demonstration.

Content
A person's goal is to bake a cake. For each of the steps, list all the state changes of involved entities and attributes. Step: Mix the eggs with flour. <generation for complete> - The shape of eggs were<generation for states> solid before and fluid after.

Table 13: Our prompt for `text-davinci-003` for both the states prediction sub-task and the complete-sentence evaluation format, followed by 1-shot demonstration.

Role	Content
User	A person's goal is to bake a cake. For each of the steps, you will list all state changes of entities and attributes. You will answer in this format: - attribute_name of entity_name was before_state before and after_state after For example: - temperature of oven was cool before and hot afterwards. Are you ready?
Assistant	Yes, I'm ready.
User	Step: Mix the eggs with flour.
Assistant	<generation for complete> - The shape of eggs were<generation for states> solid before and fluid after.

Table 14: Our prompt for `gpt-3.5-turbo` for both the states prediction sub-task and the complete-sentence evaluation format, followed by 1-shot demonstration.