The Coreference under Transformation Labeling Dataset: Entity Tracking in Procedural Texts Using Event Models

Kyeongmin Rim* and **Jingxuan Tu*** and **Bingyang Ye** and
Marc Verhagen and **Eben Holderness** and **James Pustejovsky**
Department of Computer Science
Brandeis University
Waltham, Massachusetts
{krim,jxtu,byye,verhagen,egh,jamesp}@brandeis.edu

Abstract

We demonstrate that coreference resolution in procedural texts is significantly improved when performing transformation-based entity linking prior to coreference relation identification. When events in the text introduce changes to the state of participating entities, it is often impossible to accurately link entities in anaphoric and coreference relations without an understanding of the transformations those entities undergo. We show how adding event semantics helps to better model entity coreference. We argue that all transformation predicates, not just creation verbs, introduce a new entity into the discourse, as a kind of generalized Result Role, which is typically not textually mentioned. This allows us to model procedural texts as process graphs and to compute the coreference type for any two entities in the recipe. We present our annotation methodology and the corpus generated as well as describe experiments on coreference resolution of entity mentions under a process-oriented model of events.

1 Introduction

Entity coreference resolution is a critical component for understanding most natural language text (Poesio et al., 2023; Sukthanker et al., 2020). However, when events in the text introduce changes to the state of participating entities, it is often impossible to accurately link entities in anaphoric and coreference relations without an understanding of the transformations those entities undergo. For example, events can bring about changes in entities that are not reflected in actual text mentions:

(1) a. Chop **the garlic** [WHOLE];
b. Put **it** [CHOPPED] in the pan.

That is, while **it** is *anaphorically* bound to **the garlic**, it is not strictly coreferential, as the garlic has undergone a transformation (Mitkov et al., 2000).

Events can also introduce new entities into the discourse or narrative, through the use of creation predicates (Asher, 1993; Badia and Saurí, 2000). This is pervasive in procedural text, where the goal is to describe a sequence of transformations to apply to multiple objects to build up a goal object. This can be seen, for example, in (2a), where the entities are transformed into a hidden result argument, which then licenses the definite NP *the mixture* in (2b). In addition, procedural text witnesses both *argument drop*, as in (2d), where the direct object is elided, as well as *metonymies*, where a container refers to its content, as with *bowl* in (2d).

(2) a. Mix **flour** and **water** in a bowl.
b. Set **the mixture** [FLOUR + WATER] aside.
c. Beat **the eggs**.
d. Add ∅ [BEATEN EGGS] to the bowl.

In this paper, we demonstrate how a process-oriented event model (POEM), based on Dynamic Event Structure proposed in Pustejovsky and Moszkowicz 2011; Pustejovsky 2013, motivated by and generalized from GL-VerbNet (Brown et al., 2022), can significantly help classify entity coreference in procedural texts. We argue that all transformation predicates, not just creation verbs, output a new entity into the discourse, as a kind of *Generalized Result Role*, which is typically not textually mentioned (Jezek and Melloni, 2011). This allows us to model procedural texts as input/output (I/O) process graph structures, as shown in fig. 1.

Each edge in the graph represents POEM, an event reduced to an I/O process. The "output" nodes of events are generalization of the result role from the VerbNet frames, as well as placeholders of syntactic drops and shadow arguments. The POEM graph, thus, is one way to serialize the abstraction of complex semantics including event-argument structures, subevent structures, temporal ordering and coreference chains, which we can unfold to re-construct other semantic structures.

[*] These authors contributed equally to this work.

Findings of the Association for Computational Linguistics: ACL 2023, pages 12448–12460
July 9-14, 2023 ©2023 Association for Computational Linguistics

For example, from the graph, one can compute the type of (conventional) coreference or what we call a “coreference under transformation” relation for any two entities in the recipe.

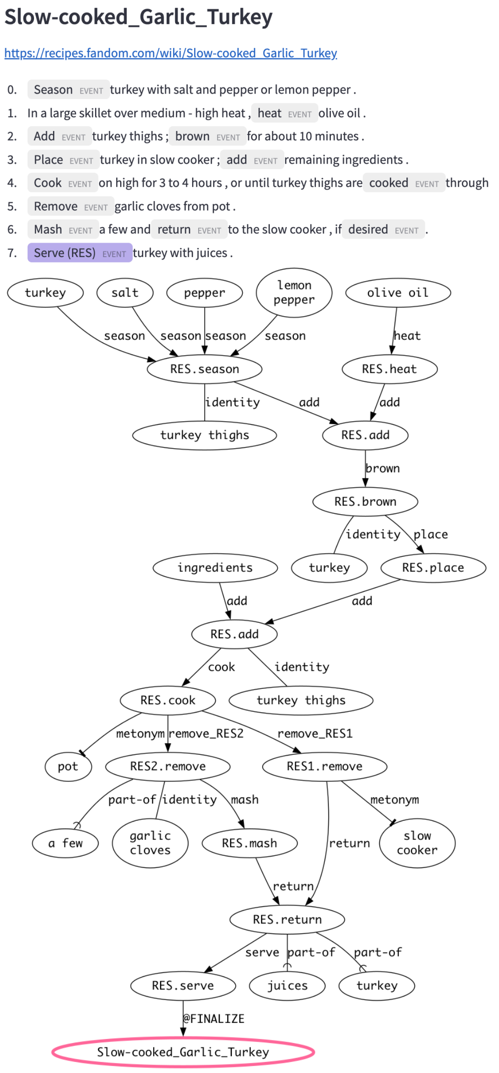
To this end, we present CUTL [1], a novel annotation methodology and dataset that integrates both the tracking of entity transformations and coreference chains of entities into a single framework. Our pilot annotation contains 100 double-annotated cooking recipes, showing high agreement on relation F1 scores. Based on our process-oriented semantic model of events, we introduce a distinction between two relations: (i) *Coreference under Identity (CuI)*, where two entities have identical state information; and (ii) *Coreference under Transformation (CuT)*, where some change has occurred distinguishing two entities.

We, then, use various methods from Tu et al. 2023 to *paraphrase* transformed entities (generalized result role) that do not appear as textual mentions, while being aware of transformations entities have undergone. We use the annotated data to train models to predict various coreference relations between entities and show the value of transformation-aware entity representation in developing a coreference resolution system that works with entities in procedural text. Our experiment also shows an interaction between our semantic model and LLMs to generate reliable and natural paraphrases.

The contributions outlined in this paper include: (1) studying the anaphoric and coreference behavior inherent in procedural texts, focusing on cooking recipes; (2) operationalization of the POEM, where steps in a procedure are annotated with explicit I/O entity nodes, regardless of whether they are mentioned in the text; (3) creation of an annotation guideline and GUI environment based on the event model, identifying events and their semantic class, all ingredient entities, and set of coreference relations between entities, typed according to the kind of transformation; and (4) the creation of a dataset, CUTL, containing these entity coreference links and the events involved.

2 Related Work

Understanding procedural narratives involves many core competencies in language comprehension (Fang et al., 2022). Not only is it crucial to perform anaphora resolution (Poesio et al., 2016), but equally important is to perform state tracking on the entities as they undergo transformations described in the text (Bosselut et al., 2017).



Slow-cooked_Garlic_Turkey

https://recipes.fandom.com/wiki/Slow-cooked_Garlic_Turkey

0. Season EVENT turkey with salt and pepper or lemon pepper .
1. In a large skillet over medium - high heat , heat EVENT olive oil .
2. Add EVENT turkey thighs ; brown EVENT for about 10 minutes .
3. Place EVENT turkey in slow cooker ; add EVENT remaining ingredients .
4. Cook EVENT on high for 3 to 4 hours , or until turkey thighs are cooked EVENT through
5. Remove EVENT garlic cloves from pot .
6. Mash EVENT a few and return EVENT to the slow cooker , if desired EVENT .
7. Serve (RES) EVENT turkey with juices .

Figure 1: A full recipe text and its Coreference under Transformation Labeling (CUTL) annotation in graph form. Nodes in the graph represent ingredients and their referring expressions. Events (gray boxes in the text) are reduced to simple I/O processes and are represented as edges in the graph.

The task of anaphora resolution covers a range of coreference relations (Poesio et al., 2023), as well as non-identity anaphoric relations, known as bridging phenomena (Clark, 1977; Asher and Lascarides, 1998). Most work on anaphora resolution has focused on declarative narratives or dialogue datasets (Pradhan et al., 2012a; Poesio et al., 2023).

Interestingly, while there are several datasets of procedural texts that have been annotated and studied, these have been mostly in the context of entity state tracking and QA tasks (Mishra et al., 2018;

[1] annotation data, scheme, tool and experiment code is available at `https://github.com/brandeis-llc/dp-cutl`

Yamakata et al., 2020; Tu et al., 2022a), rather than coreference resolution; two notable exceptions include (Mysore et al., 2019) and (Fang et al., 2022).

Examples of how entity state tracking datasets contribute to reasoning and inferencing tasks can be seen with (Bosselut et al., 2017), who presents the Neural Process Networks to model the state changes in procedural texts. The actions and entities are both predefined sets. They use soft attention to select actions and entities from the predefined sets to generate a state embedding for each entity at every step in the recipe. (Dalvi et al., 2019) is an example of how entity state tracking datasets contribute to reasoning and inferencing tasks. The paper extends the ProPara dataset (Mishra et al., 2018; Tandon et al., 2018) which contains texts describing processes. Workers were given a prompt (e.g., "What happens during photosynthesis?") and then asked to author a series of sentences describing the sequence of events in the procedure. The goal is to predict the state (location, created, destroyed) change of all the participants. Also working with ProPara, (Kazeminejad et al., 2021) approach the task of tracking state change by first parsing every sentence in ProPara with the VerbNet Parser (Gung and Palmer, 2021), and then leveraging the lexical information from VerbNet and PropBank to predict the state change.

The interaction of anaphora resolution with state tracking makes it challenging to classify the relationships that result between entities mentioned in the text, in order to judge whether they are coreferential or somehow related, but not the same. To this end, the above distinction between coreference and bridging (non-identity anaphora) becomes relevant (Hou et al., 2018). This is how Fang et al. 2022 approach the problem of NP reference in procedural text. They first adopt the distinction made in Rösiger et al. 2018 between two types of bridging (referential, where the NP requires an antecedent to be fully understood; and lexical, which may involve any number of lexical semantic relations between two NPs. Their dataset (RecipeRef) of coreference relations includes both coreference and bridging relations. For the latter, they distinguish three types, depending on the state of the entities being associated: (a) no change; (b) physical change; and (c) chemical change.

Another work focusing on anaphora in recipes is Jiang et al. 2020, which introduces RISeC, a dataset for extracting structural information and resolving zero anaphora from unstructured recipes. Our work is in the same spirit, as they utilize a general lexical resource, PropBank, rather than a limited inventory of pre-defined predicates as in Tasse and Smith 2008 . The corpus provides semantic graph annotations of (i) recipe-related entities, (ii) generic verb relations (from PropBank) connecting these entities, (iii) zero anaphora verbs having implicit arguments, and (iv) textual descriptions of those implicit arguments. The corpus however, does not contain state changes between entities.

Yamakata et al. 2020 introduce a corpus of annotated English recipes. The annotation is a flow graph (i.e., DAG with a single root) including entities and relationships between these entities. The direction of edges also indicates dependencies between actions. The label of edges explicitly specify the state change of entities. While their graph representation is similar to ours in many respects, they do not encode coreference or bridging relations.

There are newly emerging datasets focusing on both anaphora and bridging, many of them released as part of the most recent shared task on anaphora and bridging relation detection (Yu et al., 2022). Unfortunately, procedural datasets were not included in this task.

3 CUTL Dataset and Annotation Scheme

Procedural texts, such as recipes, are interesting to CL researchers for several reasons. One of those is that they are step-driven narratives requiring minimal temporal ordering recognition. As a result, semantic interpretation can focus on the changes that are taking place in the course of a sequence of events in the narrative, while assuming that the events are temporally ordered in a narrative progression. The goal of our CUTL annotation is to create a dataset of cooking recipe texts annotated with the following information:

- **Events**, typed with their semantic subclass;
- **Referring expressions** of event arguments;
- **I/O relations** between an event and its arguments (Jezek and Pustejovsky, 2019);
- **Coreference relations** between named entities in the recipe, when they exist.

The relations we adopt reflect the view laid out in Recasens et al. 2011, which distinguishes *near-identity* from (*true*) identity when drawing coreference relations between referring expressions. Thus we identify those relations derived from I/O as nearidentities and other non-I/O coreference relations as true identities.

3.1 Data Source and Mention Annotation

We reviewed publicly available recipe annotation datasets and decided to build our dataset on top of the existing R2VQ corpus (Tu et al., 2022a) from SemEval 2022, as it already contains event-structural semantic annotation layers.[2] Specifically, R2VQ has an SRL layer (SRL columns) that includes verb sense disambiguation, predicate-argument structure, and argument role labels. Additionally, it provides domain-specific "cooking entity" labeling (cooking action events, ingredients, tools, habitats) for event and entity spans (ENTITY columns). For this work, our main focus is on cooking actions, food ingredients, and their referring expressions. Thus, to generate lists of ingredients (and referring expression) mentions for the CUTL annotation, we used the union of `Patient` and `Theme` arguments from the SRL layer and `INGREDIENT` and `HABITAT` labels from the ENTITY column in R2VQ. For event mentions, we used the union of predicate spans from SRL and `EVENT` from ENTITY. To distinguish simple change of locations from entity state changes (transformations, see §3.2), we hand-labeled the change-of-location verb subclass in order to use it for relation labeling, partly adopting event subclass categories from (Im and Pustejovsky, 2010). Even though the base dataset has argument structures already annotated, because the semantics of the POEM is not directly mappable to semantics "role" names, we only took advantage of argument span annotation. The base dataset also has coreference chain annotation, but it is not compatible to this work because it did not consider near-identity. Thus we discarded the COREF column as well.

To model events as simple I/O transformation processes, our annotation scheme is pivoted on two critical assumptions: (1) textual ordering of events in a recipe reflects the temporal order of cooking actions; and (2) every event predicate has a result, regardless of whether it is mentioned in the text.

Based on the first assumption and considering document length and event number distribution, we sampled 100 recipes from the R2VQ dataset to annotate. This subset does not include any recipe that violates the temporal ordering assumption. Table 1 shows the statistics of the ingredient entities in the CUTL annotation. Compared to the original R2VQ, CUTL contains much richer hidden entity annotation from the I/O relations. Table 2 shows different types of mentions we used in the CUTL annotation.

Avg. # of entities per recipe	Explicit	Hidden
EVENT	10.6	N/A
INGREDIENT (input)	12.0	**9.4**
INGREDIENT (output)	1.0	**10.4**
R2VQ		
INGREDIENT (participant)	11.5	5.7
INGREDIENT (result)	1.1	2.5

Table 1: Statistics of cooking entity from the CUTL annotation of 100 recipes. R2VQ annotates two relations (*participant-of* and *result-of*) between the entity and the event. It can be roughly mapped to the I/O relations. However, the I/O relations has a broader coverage of the hidden entities.

Mention	**Examples**
Event	cut, slice, bake, peel, ...
C.Loc event	throw, put, pour, ...
Location	pot, skillet, oven, board, ...
Ingredient	beef, onion, salt, water, ...
Result states	soup, dough, pizza, mixture, ...
Pronouns	it, them, half, ...
Property (shape, size, ...)	Roll dough into [balls] Cut into [2-inch pieces]

Table 2: Types of mentions of interest in the CUTL annotation.

3.2 Coreference Relation Annotation

One of the key goals of the annotation task is to identify three types of event-structural information in the text that, together form the fundamental building blocks of the POEM : (1) EVENT PREDICATES, (2) INPUT ENTITIES, (3) RESULT/OUTPUT ENTITIES. For cooking events "input"s are naturally understandable as the ingredients used for an action. Syntactically, we treat all the objects of an event predicate as its inputs (although, often they are hidden from the surface form as we saw in examples in the section 1. Thus, in a sense, an input and the output of an event are coreferential, only considering the transformation that the input underwent during the event. We call this relation **Coreference under Transformation (CuT)**.

The innovative aspect to our model being assumed here is that every event must have one or more result entities, whether they are explicitly

[2] R2VQ data and its specification was obtained from the shared task's website.

mentioned in the text or not. Compare the recipe steps in example (3) below.

(3) a. [**Form**$_{evt}$] the mixture into [**patties**$_{res}$].
b. [**Mix**$_{evt}$] flour and water [$\varnothing_{res}$].
c. [**Remove**$_{evt}$] [**skin**$_{res1}$] and [**bones**$_{res2}$] from the halibut. [$\varnothing_{res3}$].

In (3a), we get a physically re-shaped meat mixture as the result of the action, and [**the mixture**$_{ent}$] and [**patties**$_{ent}$] are coreferential under the [**form**$_{evt}$] transformation. In (3b), we have two inputs and an aggregated object as the result. Because the result is hidden, there is no token we can directly anchor the mixture to, which we deal with by re-using the event predicate span as the anchor for the result, creating a *phantom* entity (indicated by RES. prefix below) referring to the output of the transformation. The same applies to the separation process in (3c), which is different from the others in that it results in multiple outputs. Example (3′) shows CuT relations from 3.

(3′) a. [**mixture**$_{ent}$] $\xrightarrow[\text{TRANSFORMATION}]{\text{form}}$ [**patties**$_{ent}$]
b. [**flour**$_{ent}$] $\xrightarrow[\text{AGGREGATION}]{\text{mix}}$ [**RES.mix**$_{ent}$]
[**water**$_{ent}$] $\xrightarrow[\text{AGGREGATION}]{\text{mix}}$ [**RES.mix**$_{ent}$]
c. [**halibut**$_{ent}$] $\xrightarrow[\text{SEPARATION}]{\text{remove}}$ [**skin**$_{ent}$]
[**halibut**$_{ent}$] $\xrightarrow[\text{SEPARATION}]{\text{remove}}$ [**bones**$_{ent}$]
[**halibut**$_{ent}$] $\xrightarrow[\text{SEPARATION}]{\text{remove}}$ [**RES3.remove**$_{ent}$]

The advantage of using these *phantom* spans is twofold; (i) we can directly draw a relation between the input and the output or between a new name and a non-mention output (when *redescription* (Badia and Saurí, 2000) happens in the text); and (ii) when a following event takes a result of the current event as an input, we can pass the newly created phantom node. Example (2′) is the set of coreferences from example (2), showing how phantom spans are used.

(2′) a. [**flour**$_{ent}$] $\xrightarrow[\text{AGGREGATION}]{\text{mix}}$ [**RES.mix**$_{ent}$]
[**water**$_{ent}$] $\xrightarrow[\text{AGGREGATION}]{\text{mix}}$ [**RES.mix**$_{ent}$]
b. [**RES.mix**$_{ent}$] $\underset{\text{REDESCRIPTION}}{=}$ [**mixture**$_{ent}$]
[**mixture**$_{ent}$] $\underset{\text{CHANGE-OF-LOCATION}}{\overset{\text{set}}{=}}$ [**RES.set**$_{ent}$]
c. [**the eggs**$_{ent}$] $\xrightarrow[\text{TRANSFORMATION}]{\text{beat}}$ [**RES.beat**$_{ent}$]
d. [**RES.beat**$_{ent}$] $\xrightarrow[\text{AGGREGATION}]{\text{add}}$ [**RES.add**$_{ent}$]
[**RES.set**$_{ent}$] $\xrightarrow[\text{AGGREGATION}]{\text{add}}$ [**RES.add**$_{ent}$]

Fang et al. 2022 attempt to work around these issues by treating CuTs as *bridging* relations to an input entity, but only when the output is “redescribed” as a text mention. We believe we should avoid using the term “bridging” too liberally for these cases. Furthermore, when the redescription occurs less frequently (only after several transformations), it will identify long-distance bridging relations that require cognitive jumps in the annotators’ mind, which is not necessarily recorded in the annotation data.

We distinguish the CuT relation from **Coreference under Identity (CuI)**, which is the more conventional definition of coreference, and some of bridging relations such as part-whole relation. In addition to one-to-one IDENTITY relations including anaphoric pronouns, we also annotated locational METONYMY and MERONYMIC relations as subtypes of CuI. As discussed earlier, annotators are presented with automatically generated phantom result entities for every event predicate. So the redescription operation is identified as a CuI link in the annotation environment. One note to make here is that when an event predicate falls under the CHANGE-OF-LOCATION semantic subclass (Tu et al., 2022b) and the I/O annotation is single-in and single-out, we call this relation between the input and the output as a CuI even though the relation is mediated by an event, as the only difference the event made is the location of the entity, thus not transformation.

In summary, we used the following typology of coreferences to link two entities. Some are direct links between two entities while the others are mediated by events under the transformation.

COREFERENCE UNDER IDENTITY

1. Identity: strict coreference of two entities.
2. Meronymy: relation between two entities when one end is referring to an inseparate part of the other.
3. Metonymy: links between an ingredient entity and a location entity when the location entity is used as a container for the food. [3]
4. Change of location: single-in, single-out under CHANGE-OF-LOCATION transformation.[3]

COREFERENCE UNDER TRANSFORMATION

[3] These sub-categories of relations are not annotated by the annotator, but automatically inferred from the structural information or pre-compile R2VQ annotation. Annotators still need to draw a link between two entities, but, for example, when one end is from HABITAT annotation, the relation label is automatically switched to `metonym`. Or when an annotator draws a link between a single input entity and an event, `transformation` label is used. However, if the event predicate is pre-annotated as a CHANGE-OF-LOCATION event subclass, the label will be `identity` instead.

1. Transformation: a one-to-one link between the input node and the output node of a transformation event.[3]
2. Aggregation: a many-to-one link from input nodes to an output node.[3]
3. Separation: a one-to-many link from an input node to output nodes

Annotations are encoded as a directed acyclic graph where (1) leaves are primitives (base ingredients), (2) the root is the title of the recipe, corresponding to the final state in the graph, (3) edges represent coreference relations and (4) internal nodes correspond to inputs and outputs of events – these are phantom entities, some of which are linked to redescription nominals via CuI annotation.

3.3 The CUTLER Annotation Environment

We developed a GUI annotation environment, CUTLER. It uses a simple table-based click-only workflow to quickly mark inputs and outputs of an event, types of the event, and coreference groups among entities. Figure 2 shows a screenshot of the CUTLER interface with a quick description of the annotation workflow. We believe the conceptually simple and streamlined interface of the CUTLER annotation environment significantly reduced annotator cognitive load, resulting in improved annotation speed and high inter-annotator agreement. The full guidelines for the CUTL annotation and the CUTLER software are available under open-source licenses in the data and code repository of the work.

3.4 Inter-Annotator Agreement and Gold Standard Dataset

Annotation of the 100 recipes was done in 4 rounds by 7 researchers and graduate students from the linguistics and computer science departments of a US-based university. Each document was dually annotated and Inter-Annotator Agreement (IAA) was computed at the end of each round. Pairs of annotators then met to adjudicate disagreements and create a finalized gold standard annotation. We used pairwise F1 as our primary IAA metric, which was uniformly high across labels, rounds, and annotator pairs with a mean $F1 = 86.9$. Metonymy and meronymy relations constituted the labels with the highest disagreement. This is partially due to having the fewest instances in the dataset, as well as the inherent ambiguity in each of these labels; during adjudication it was often found that both annotations were semantically valid. Encouragingly, CuT-related labels – the primary focus of this work – had consistently high agreement ($F1 > 90.0$ in the majority of documents).

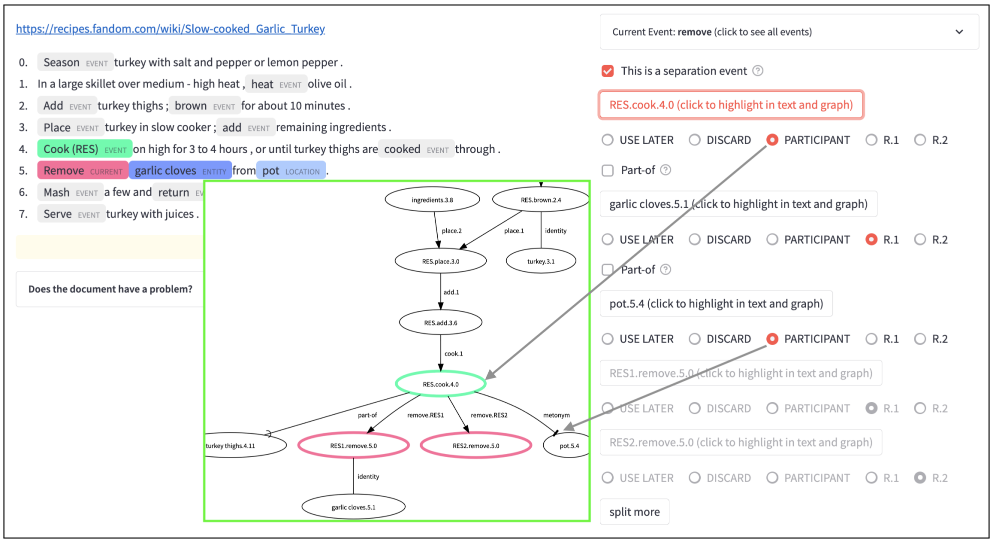
4 Coreference resolution with CuT

We implemented a coreference resolution system using CUTL dataset. This section will describe the system design and the performance of the system.

Experiment Setup Under the POEM and CuT relations, coreference "chains" now can include phantom entity mentions (with RES. prefix). These phantom mentions serve two purposes; 1) make all event outputs explicit, 2) fill syntactic drop arguments in the following event. However, these mentions textually do not exist in the text, thus cannot be easily modelled by any language model-based system, that are based on vector embedding of the surface text. To address this problem, we adopted Dense Paraphrasing (DP), a text enrichment technique (Tu et al., 2023) to first recover all drop argument (as empty slots) and "paraphrase" drop argument nodes and RES. nodes, to create natural language representation of the CUTL annotated data. Concretely, we apply both PREFIXP and SUBGRAPH-GPT methods from Tu et al. 2023 to all the drop and phantom entities from the recipe to generate paraphrases. PREFIXP is a heuristic method that paraphrase the entities by prepending the prefixes to reflect changes due to actions. SUBGRAPH-GPT use the GPT-3 model (Brown et al., 2020) to paraphrase the linearized subgraph that is rooted from the drop argument node or the RES. node. Figure 3 shows an example from the different paraphrasing methods.

Once the text is replaced with paraphrase with recovery of of drop arguments and insertion of generalized result nodes, we can use the new text in a coreference resolution task.[4] For the coreference resolution task, we adopt the neural coreference model and the configuration from (Fang et al., 2021) and formulate the problem into a joint training of an antecedent assignment model (Lee et al., 2017) and a classification model. The system first detects all possible mentions of coreference. Then, for CuI resolution, the coreference resolution model would learn to assign a set of antecedents to each mention. And for CuT resolution, the bridg-

[4] We format the task input sentences with additional paraphrases based on simple heuristics: `[To get Z], do X, [Y]`. Brackets represent the text that is inserted.



https://recipes.fandom.com/wiki/Slow-cooked_Garlic_Turkey

0. Season EVENT turkey with salt and pepper or lemon pepper .
1. In a large skillet over medium - high heat , heat EVENT olive oil .
2. Add EVENT turkey thighs ; brown EVENT for about 10 minutes .
3. Place EVENT turkey in slow cooker ; add EVENT remaining ingredients .
4. Cook (RES) EVENT on high for 3 to 4 hours , or until turkey thighs are cooked EVENT through .
5. Remove CURRENT garlic cloves ENTITY from pot LOCATION .
6. Mash EVENT a few and return EV
7. Serve EVENT turkey with juices .

Does the document have a problem?

Current Event: **remove** (click to see all events)

☑ This is a separation event

RES.cook.4.0 (click to highlight in text and graph)

○ USE LATER ○ DISCARD ● PARTICIPANT ○ R.1 ○ R.2

□ Part-of

garlic cloves.5.1 (click to highlight in text and graph)

○ USE LATER ○ DISCARD ○ PARTICIPANT ● R.1 ○ R.2

□ Part-of

pot.5.4 (click to highlight in text and graph)

○ USE LATER ○ DISCARD ● PARTICIPANT ○ R.1 ○ R.2

RES1.remove.5.0 (click to highlight in text and graph)

○ USE LATER ○ DISCARD ○ PARTICIPANT ● R.1 ○ R.2

RES2.remove.5.0 (click to highlight in text and graph)

○ USE LATER ○ DISCARD ○ PARTICIPANT ○ R.1 ● R.2

split more

Figure 2: CUTLER in action. By providing step-by-step annotation tables and real-time graph generation, annotators can focus on local changes while keeping track of entity states and global changes by looking at the graph (green box). Available entities are presented as blue spans (left) with radio options to specify their relation (right). Annotators draw relations between all available entities and the current event (pink span in the left, pink oval in the rendered graph). When multiple entities are linked as the same relation (by putting on the same 'column' in the radio option table), a CuI label is inferred between them based on the entity types, the event type, and current I/O structure.

Sentence:
..., bring peas and water to boil and reduce to simmer .. ⟶ `RES.simmer`

PrefixP:
`RES.simmer` ⟶ `'simmered boiled peas and water'`

Subgraph-GPT:

```
                ['simmer',
RES.simmer ⟶   [['boil',                      ⟶  'pea soup'
                  ['peas', 'water']]]]
```

Figure 3: Paraphrase of the the RES. node RES.SIMMER from the PREFIXP and SUBGRAPH-GPT methods. SUBGRAPH-GPT first extracts the linearized graph rooted from RES.SIMMER, and then generate the final paraphrase with GPT-3.

ing model will do a multi-class classification on each pair of mentions detected.

Given the hierarchical nature of our CUTL types, we design the coreference resolution in two fashions: (i) Coarse: we only consider if there is transformation due to event, so there will only be CuI and CuT, and we treat all sub-relations under CuI as coreference. (ii) Fine-grained: we consider each relation type as an individual class and only treat Identity as coreference.

Machine learning model details We train a neural coreference resolution model with a configuration similar to (Fang et al., 2022; Lee et al., 2018). Specifically, we use 300-dimensional GloVe embeddings (Pennington et al., 2014) with window size=2 for head word embeddings. And we train ELMo embeddings (Peters et al., 2018) on both CUTL and RecipeRef corpora. We also trained a CNN model with windows of 3, 4, and 5 to learn the character embeddings, and concatenate all three embeddings as the token representation. For each experiment, we do a 5-fold cross validation and train the model for 20 epochs on 4 NVIDIA Titan Xp GPUs.

Results Since our data contains one-to-many coreferential relation (SEPARATION) and many-to-one relation (AGGREGATION), the traditional coreference resolution metrics (Pradhan et al., 2012b) are not suitable for our task, and we evaluate our experiments using F1. Table 3 shows the results of our experiments on 5-fold cross validation in both coarse and fine-grained ways. It is not surprising to see that it is more difficult to do coreference resolution with a more complex set of relation types, given the results of the coarse setting on both inputs are higher than the fine-grained. It also shows that the results from GPT-base paraphrases as inputs are higher than the inputs using DENSE paraphrasing on both overall and most of the fine-grained relations. Table 4 breaks down the results into the fine-grained coreferential relations. On each relation type, we evaluate using MUC, BCUBED and CEAF F1 from (Pradhan et al., 2012b) and their

		Coreference			CUT		
Setting	Input	Avg.P	Avg.R	Avg.F1	Avg.P	Avg.R	Avg.F1
Coarse	PrefixP	82.46 (±5.31)	9.31 (±6.81)	16.73 (±6.09)	86.05 (±1.92)	46.41 (±5.06)	60.29 (±4.01)
	Subgraph-GPT	85.68 (±9.81)	11.02 (±3.50)	**19.07** (±5.67)	88.12 (±3.18)	47.25 (±2.84)	**61.09** (±2.88)
Fine	PrefixP	87.28 (±6.36)	11.60 (±0.83)	20.02 (±1.38)	85.19 (±1.10)	41.15 (±1.59)	54.89 (±1.30)
	Subgraph-GPT	89.57 (±4.37)	11.67 (±1.86)	**20.11** (±2.92)	82.99 (±2.10)	44.50 (±2.72)	**57.33** (±1.95)

Table 3: Coreference resolution results on 5-fold cross validation.

	PrefixP					Subgraph-GPT				
Relation	F1	MUC-F1	BCUBED-F1	CEAF-F1	AVG.F1	F1	MUC-F1	BCUBED-F1	CEAF-F1	AVG.F1
Identity	56.80	58.22	11.91	11.88	27.34	**58.01**	78.84	19.14	18.62	**38.87**
Metonym	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Meronym	**25.00**	16.57	3.45	12.40	**10.80**	16.00	3.88	0.16	3.25	2.43
Trans.	64.92	74.71	22.78	38.80	45.43	**65.52**	87.02	25.67	28.52	**47.07**
Agg.	74.39	83.91	22.31	31.97	46.06	**76.72**	89.41	25.59	33.25	**49.42**
Sep.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 4: Coreference resolution results of fine-grained relations on the hold out test set.

average values. We also include the F1 scores. The observed outcomes align with the overall performance presented in table 3. It is noteworthy that, with the exception of the Meronym relation, utilizing the GPT-base paragraphs as input yields higher F1 scores compared to the Dense paragraphs. This finding further supports the notion that the GPT-base paragraphs exhibit improved performance in coreference resolution.

5 Discussion and future work

5.1 Measuring agreement of CUTL

To compute IAA of the CUTL annotation (§3.4), we tried different metrics including naïve Kappa score and CoNLL coreference score, but decided to go with only F1. We view our research problem differently from traditional coreference tasks in two ways: (1) We have multiple coreferential types. This results in one entity being in more than one coreferential chain. (2) In "separation" relation, an arbitrary number of new hidden arguments can be added which means the set of entities is not fixed. Traditional metrics like Kappa or CoNLL can only measure one aspect of randomness of our data, while F1 can show the agreement in a more all-around fashion. The same logic applies to reporting of our experiment. But it would be an interesting topic to develop new metrics or re-formulate our linking task into a labeling task compatible with Kappa-family metrics.

5.2 Data selection and limitations

The annotation scheme proposed in this work is designed to focus on non-identidy coreference, CuT, and is not able to handle some complex linguistics phenomena. That includes (not limited to) complex temporal ordering, VP or NP ellipsis under conjunction and/or disjunction, event negation. As a result, during data selection process, we had to look for those linguistic features and excluded documents with them from the data set.

Specifically, to limit the scope of the research, we intentionally limited our analysis to data that:

- Is temporally linear
- Has a single terminal state
- Has a high density of object transformations referred to explicitly throughout the text

We chose to work within the cooking recipe domain because it easily satisfies criteria. However, procedural text in general satisfies these three conditions, and our current model is therefore compatible with a broader range of domains than strictly recipes. In future work, we intend to broaden the scope to include more varied domains, such as news data and narratives.

During the manual curation of 100-document subset, we did not encounter any annotation of nominal events, and therefore this work ipso facto involves only events extracted from verbs. Although event recognition is not the primary research focus of this work, being able to additionally identify different types of lexical trigger of events is indeed important when considering broader domains. We plan to integrate our framework with other lexical resources in the future, and event recognition will receive more focus.

5.3 Event semantics

For this study, we directly adopted (Tu et al., 2022b) and used simple three-way subclass categorization for event semantics. In the future, we will make a finer event type categorization utilizing existing large lexical resources such as GL-VerbNet (Brown et al., 2022). We hypothesize that utilizing finer and semantically loaded event subclasses will help empirical investigations of nominal redescriptions as well as improve automatic paragraph generation.

6 Conclusion

In this paper, we presented a new dataset, CUTL, annotated using a novel integration of integrating event semantics and coreference linking annotation. We applied a process-oriented event model and argument structure as coreference relations between event input(s) and output(s). We showed that using CUTL is a very efficient way of analyzing and annotating entity transformation and coreference chains in procedural text, by conducting pilot annotations on cooking recipe text. The CUTL dataset and annotation material are available under open-source licenses. Additionally, we conducted multi-stage experiments to build the baselines for coreference identifier and classifier that focus on utilizing our human annotations. The results from the coreference resolution systems show that the subgraph representation of our annotation is a good resource for LLMs such as GPT-3 to generate reliable paraphrases in natural language, which can further improve the multi-class coreference resolution task.

References

Nicholas Asher. 1993. *Reference to abstract objects in discourse*, volume 50. Springer Science & Business Media.

Nicholas Asher and Alex Lascarides. 1998. Bridging. *Journal of Semantics*, 15(1):83–113.

Toni Badia and Roser Saurí. 2000. Enlarging hpsg with lexical semantics. In *Proceedings of CICLing*, pages 101–122.

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*.

Susan Windisch Brown, Julia Bonn, Ghazaleh Kazeminejad, Annie Zaenen, James Pustejovsky, and Martha Palmer. 2022. Semantic representations for nlp using verbnet and the generative lexicon. *Frontiers in artificial intelligence*, 5.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Herb Clark. 1977. Bridging. *Thinking: Readings in Cognitive Science*.

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.

Biaoyan Fang, Timothy Baldwin, and Karin Verspoor. 2022. What does it take to bake a cake? the reciperef corpus and anaphora resolution in procedural text. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3481–3495.

Biaoyan Fang, Christian Druckenbrodt, Saber A Akhondi, Jiayuan He, Timothy Baldwin, and Karin Verspoor. 2021. ChEMU-ref: A corpus for modeling anaphora resolution in the chemical domain. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1362–1375, Online. Association for Computational Linguistics.

James Gung and Martha Palmer. 2021. Predicate representations and polysemy in verbnet semantic parsing. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 51–62.

Yufang Hou, Katja Markert, and Michael Strube. 2018. Unrestricted bridging resolution. *Computational Linguistics*, 44(2):237–284.

Seohyun Im and James Pustejovsky. 2010. Annotating lexically entailed subevents for textual inference tasks. In *Twenty-third international flairs conference*.

Elisabetta Jezek and Chiara Melloni. 2011. Nominals, polysemy, and co-predication. *Journal of cognitive science*, 12(1):1–31.

Elisabetta Jezek and James Pustejovsky. 2019. Dynamic interpretation of predicate-argument structure. *Lingue e linguaggio*, 18(2):179–208.

Yiwei Jiang, Klim Zaporojets, Johannes Deleu, Thomas Demeester, and Chris Develder. 2020. Recipe instruction semantics corpus (risec): Resolving semantic structure and zero anaphora in recipes. In *AACL-IJCNLP 2020, the 1st Conference of the Asia-Pacific Chapter of the Association Computational Linguistics and 10th International Joint Conference on Natural Language Processing*, pages 821–826. Association for Computational Linguistics (ACL).

Ghazaleh Kazeminejad, Martha Palmer, Tao Li, and Vivek Srikumar. 2021. Automatic entity state annotation using the verbnet semantic parser. In *Proceedings of The Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 123–132.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Bhavana Dalvi Mishra, 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. *arXiv preprint arXiv:1805.06975*.

Ruslan Mitkov, Richard Evans, Constantin Orasan, Catalina Barbu, Lisa Jones, and Violeta Sotirova. 2000. Coreference and anaphora: developing annotating tools, annotated resources and annotation strategies. In *Proceedings of the Discourse, Anaphora and Reference Resolution Conference (DAARC2000)*, pages 49–58.

Sheshera Mysore, Zach 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. *arXiv preprint arXiv:1905.06939*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Massimo Poesio, Roland Stuckardt, and Yannick Versley. 2016. *Anaphora resolution*. Springer.

Massimo Poesio, Juntao Yu, Silviu Paun, Abdulrahman Aloraini, Pengcheng Lu, Janosch Haber, and Derya Cokal. 2023. Computational models of anaphora. *Annual Review of Linguistics*, 9.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012a. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 1–40.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012b. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

James Pustejovsky. 2013. Dynamic event structure and habitat theory. In *Proceedings of the 6th International Conference on Generative Approaches to the Lexicon (GL2013)*, pages 1–10, Pisa, Italy. Association for Computational Linguistics.

James Pustejovsky and Jessica L Moszkowicz. 2011. The qualitative spatial dynamics of motion in language. *Spatial Cognition & Computation*, 11(1):15–44.

Marta Recasens, Eduard Hovy, and M Antònia Martí. 2011. Identity, non-identity, and near-identity: Addressing the complexity of coreference. *Lingua*, 121(6):1138–1152.

Ina Rösiger, Arndt Riester, and Jonas Kuhn. 2018. Bridging resolution: Task definition, corpus resources and rule-based experiments. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3516–3528.

Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. 2020. Anaphora and coreference resolution: A review. *Information Fusion*, 59:139–162.

Niket Tandon, Bhavana Dalvi, Joel Grus, Wen-tau Yih, Antoine Bosselut, and Peter Clark. 2018. Reasoning about actions and state changes by injecting commonsense knowledge. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 57–66, Brussels, Belgium. Association for Computational Linguistics.

Dan Tasse and Noah A Smith. 2008. Sour cream: Toward semantic processing of recipes. *Carnegie Mellon University, Pittsburgh, Tech. Rep. CMU-LTI-08-005*.

Jingxuan Tu, Eben Holderness, Marco Maru, Simone Conia, Kyeongmin Rim, Kelley Lynch, Richard Brutti, Roberto Navigli, and James Pustejovsky. 2022a. SemEval-2022 Task 9: R2VQ – Competence-based Multimodal Question Answering. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1244–1255, Seattle, United States. Association for Computational Linguistics.

Jingxuan Tu, Kyeongmin Rim, Eben Holderness, Bingyang Ye, and James Pustejovsky. 2023. Dense paraphrasing for textual enrichment. In *Proceedings of the 15th International Conference on Computational Semantics (IWCS)*, Nancy, France. Association for Computational Linguistics.

Jingxuan Tu, Kyeongmin Rim, and James Pustejovsky. 2022b. Competence-based question generation. In *International Conference on Computational Linguistics*.

Yoko Yamakata, Shinsuke Mori, and John A Carroll. 2020. English recipe flow graph corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5187–5194.

Juntao Yu, Sopan Khosla, Ramesh Manuvinakurike, Lori Levin, Vincent Ng, Massimo Poesio, Michael Strube, and Carolyn Rosé. 2022. The codi-crac 2022 shared task on anaphora, bridging, and discourse deixis in dialogue. In *Proceedings of the CODI-CRAC 2022 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue*, pages 1–14.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
5

☒ A2. Did you discuss any potential risks of your work?
Our work is creating a new dataset using open-licensed (CC) recipe text describing diverse cuisine and food cultures. We don't believe that there's an eminent risk from the linguistic annotation of diverse but neutral text.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3

☑ B1. Did you cite the creators of artifacts you used?
3

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
3

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
3

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The raw data didn't include any PII

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3,5

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3,4

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
3

☒ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
annotation work is ongoing and guildeline is being updated iteratively. Tool screenshot and workflow description is provided in the paper (sec3)

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
3

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
We used existing dataset to start annotation, hence not directly collected raw data.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
3