Integrated Annotation of Event Structure, Object States, and Entity Coreference

Kyeongmin Rim and **James Pustejovsky**
Department of Computer Science
Brandeis University
Waltham, Massachusetts
{krim,jamesp}@brandeis.edu

Abstract

Understanding coreference and anaphora is still considered as a hard problem for NLP applications. Recent studies on modeling and annotating coreference and/or anaphoric relations show that the problem is a hard problem even for human expert annotators. In this work, we demonstrate an annotation environment that enables quick and easy, but still flexible annotation of coreference relations based on event semantics and argument structure, and constraints arsing from temporal logic. The main focus of the environment is to integrate annotation of lexically anchored entity state change tracking and coreference chains along the event-based entity transformation. The scheme and environment is developed as open source, and is publicly available.

1 Introduction

Coreference is linguistic phenomenon in which two or more expressions refer to a single real-world entity. Understanding coreferent relations in documents and dialogue is important in natural language processing (NLP) systems because it allows not only understanding the meaning of language, but also re-grouping events and statements around different participating entities that can be used in automatic summarization, for example.

Although annotating coreference in textual data has been an active research topic for a long time, early compilations of large corpora for computational operationalization of coreference resolution, such as MUC (Grishman and Sundheim, 1996; Hirschman and Chinchor, 1998), ACE (Doddington et al., 2004), or OntoNotes (Hovy et al., 2006; Pradhan et al., 2011), were focused on developing straightforward schemes to recognize fully identical denotations of entity mentions, especially pronominal anaphora.

However, it is often difficult to strictly define "identity" relations between two referring expressions or the denotations of those expressions. For example, in procedural texts such as cooking recipes, when entities in the text undergo a series of events that cause changes in their state, it is often impossible to accurately link entity mentions in anaphoric and/or coreference relations without modeling the differences (*state-wise*) between the "same" (*substance-wise*) entity before and after the transformations caused by events.

For example, let's consider a recipe for a PB&J sandwich where the entity "peanut butter" is mentioned multiple times. The peanut butter that is mentioned at the beginning of the recipe is the same physical substance that is mentioned at the end of the recipe. However, they are not exactly the same in that the peanut butter at the beginning of the recipe is probably in a jar, while the peanut butter at the end of the recipe is spread on bread. To accurately link these two mentions of "peanut butter" as *coreferent*, we need to model the difference between the two states of the peanut butter.

This is just one example of how difficult it can be to define coreference relations based on binary identity/non-identity classification between two referring expressions. In general, it is a challenging task that requires careful consideration of the context in which the expressions are used and commonsense knowledge of object interactions.

In this work, we demonstrate an annotation environment that can integrate annotation of temporal ordering and dependency of events, event argument structures, and coreference relations with full- and near-identity.

2 Background

Many of early research effort on coreference and anaphora annotation has been centered around identifying full identity relations. However, this approach sometimes fails to provide a rigorous definition of the *sameness* (Poesio et al., 2006) or misses many other important types of coreference relations (Zeldes, 2022), such as when two referring

Proceedings of the Sixth Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2023), pages 71–77
December 6–7, 2023

expressions refer to two different states of the same entity (Rim et al., 2023).

When it comes to technical aspects of annotating coreference, due to the highly complex nature of coreference and anaphoric relations and lack of complete one-key definition of those relations, the annotation is usually done by trained linguistic experts to create large-scale public datasets (Pradhan et al., 2012; Uryupina et al., 2016). Recently, more efforts on gamifying the coreference annotation (Chamberlain et al., 2016) or crowd-sourcing it (Gupta et al., 2023) were reported. Still, due to the fundamental complexity of the phenomena that often requires long-distance context and inevitable ambiguity by polysemous use of language, deconstructing coreference annotation tasks into crowd-friendly simple questions remains an unresolved problem. Because of reliance on highly trained expert annotators, many annotation environments specifically developed for coreference annotation are often designed to rely on heavy cognitive work of annotators. For instance, annotating coreference relations are frequently done (simultaneously with detecting entity mentions) as drawing *chains* of coreferences across different parts of a document. Thus, annotators are required to look at the entire document all the time jumping top to bottom, and use pointer devices to precisely drag-and-draw links that often graphically rendered as lines/arrows (Müller and Strube, 2006; Widlöcher and Mathet, 2012) or color-coded bag-of-mentions (Oberle, 2018; Reiter, 2018; Aralikatte and Søgaard, 2020).

More recently, identifying and modeling different types of coreference relations beyond full identity-based binary classification has been attracting more attention in the community (Recasens et al., 2010; Fang et al., 2022) These non-identity or near-identity coreference relations are often called *bridging* relations (Poesio and Artstein, 2008; Roesiger et al., 2018). More specifically, coreference study in procedural text is gaining more attention, based on corpus from cooking recipes or how-to domains (Mori et al., 2014; Prange et al., 2019; Fang et al., 2022).

Procedural texts are also a good source material for entity state tracking. This is because they often describe the steps involved in completing a task, which can be used to track the state of entities as they move through the process. Tracking the state of the entities in a procedural text is also

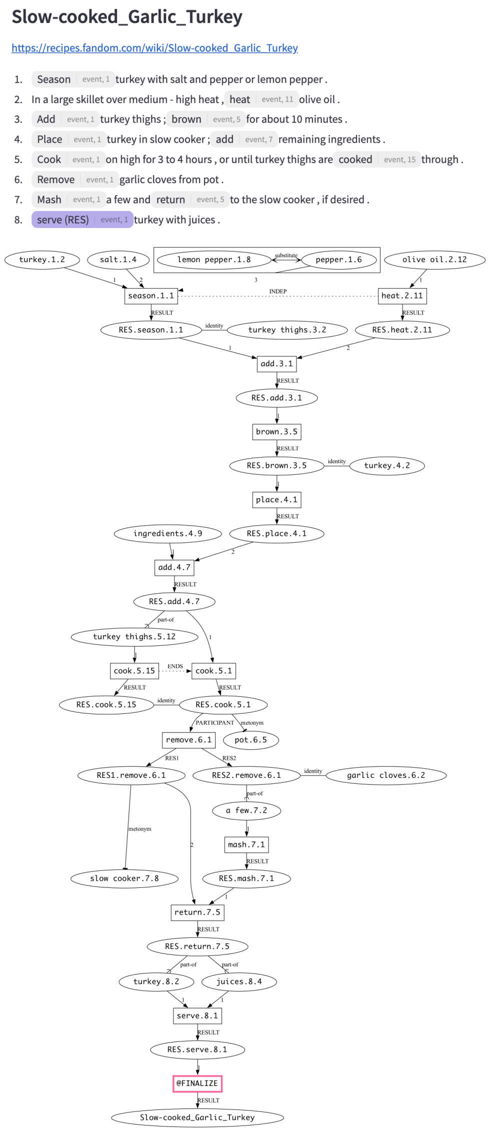

Figure 1: A full recipe text and its CUTL annotation in graph form, annotated using the new CUTLER. Rectangular nodes are events (processes) and oval nodes are entity mentions (inputs and outputs), indexed with token location. Then, coreference relations in solid edges, state changes in arrows, and event dependencies in dotted edges.

very important to understand the text and how the task is being completed. However, many studies on state change annotations have been too restrictive in terms of the state vocabulary to capture the full range of object transformations (Dalvi et al., 2018), or too open to the extent that some of the annotated data is totally unbound from any lexical clues (Tandon et al., 2020), that can cause a system based on such annotated training data to malfunction, such as hallucination in text generation (Wu et al., 2023).

3 Proposed Annotation scheme

In our previous work (Rim et al., 2023), we developed Coreference under Transformation Labeling (CUTL) annotation scheme and CUTLER[1], its paired integrated annotation environment, that enables annotation of event argument structures and coreference relations. In this work, we continue the work and propose a newer version that integrates temporal ordering of events and event dependencies in a multi-pass workflow. The annotation results are stored in graphs and thus, intermediate annotation progress can be easily visualized in real-time so that annotators can visually keep track of event-event relations (temporal order, conditional dependency), entity-event relations (argument structure), and entity-entity relations (different types of coreference relations). Our new contribution is expanding the previous work by adding annotation of event-event relations and flexibility in annotating sub-types of coreference relations.

3.1 Process-oriented event model and event dependency

The work is based on the process-oriented event model. The model is a way of representing an event as a transformation process that has inputs and outputs. Based on the model, all event mentions (verbs) create "phantom" result entities that can be used as regular entities for anchoring coreference link annotations in the rest of the timeline of the document. And the events themselves are used as a coreference relation to represent a type of near-identity between two nodes in the I/O graph. For example:

(1) a. **[Chop]**$_{res1}$ **[onion]**$_{ent1}$.
$ent1$: "(whole) `onion`"
$res1$: "(chopped onion)"
$ent1 \xleftrightarrow[\text{NEAR-IDENTITY}]{} res1$

b. **[Chop]**$_{res1}$ **[onion]**$_{ent1}$, and **[add]**$_{res2}$ **[onion]**$_{ent2}$ to the pan.
$ent1$: "(whole) `onion`"
$res1$: "(chopped onion)"
$ent2$: "(chopped) `onion`"
$ent1 \xleftrightarrow[\text{NEAR-IDENTITY}]{} res1$
$res1 \xleftrightarrow[\text{FULL-IDENTITY}]{} ent2$

When an entity undergoes multiple transformations in many steps, all the state changes are recorded as a sequence of transformations that are completely anchored on textual mentions (verbs).

However, the original annotation scheme fails to address complex temporal ordering of events and temporally conditioned event dependencies. This means that the original annotation scheme cannot take into account the fact that events can happen in a different order than they are written in the source text. Therefore, annotation is done under the assumption that all events are already temporally ordered in the text, and all source texts with complex event orders are deliberately excluded from annotation.

Since all transformation processes will take time to accomplish their goal status, we argue that temporality is an important factor to consider to understand object state changes. Furthermore, we see that some temporal relations between events are working as conditional dependencies between the events. Therefore, it is even more important to precisely model the temporal and conditional relations between events. To address this problem, we implement an annotation workflow to handle a simplified interval-based temporal logic as conditional dependencies for initiation and termination of events. This example shows how the text order and the temporal order of events can differ.

(2) a. **[Shred]**$_{evt}$ the cabbage fine.
b. **[Cook]**$_{evt}$ in butter **[melted]**$_{evt}$ over low heat until **[limp]**$_{evt}$.

- text order: *shred* → *cook* → *melt* → *(be) limp*
- temporal order:
  - (*shred*, `before`, *cook*)
  - (*shred*, `independent`, *melt*)
  - (*cook*, `begun_by`, *melt*)
  - (*cook*, `ended_by`, *(be) limp*)

3.2 Sub-types of Coreference

We proposed four sub-types of coreference-under-identity in the previous work: identity, meronymy, metonymy, and change of location. However, a widely adopted set of coreference relation subtypes does not exist. Nevertheless, there is some level of consensus in near-identity studies that the degree of sameness/difference can be measured. Based on our findings, the new version of the environment is customizable with any identity-based

[1] `https://github.com/brandeis-llc/dp-cutl`

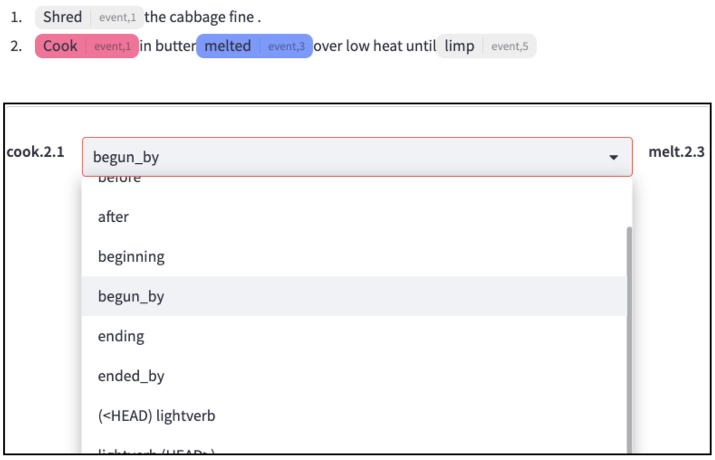

Figure 2: Event relation annotation step in CUTLER.

coreference sub-types, instead of hard-coding the relations labels into the annotation environment. New implementation enables annotation task designers to add new sub-types of coreference relations while keeping partial or total order between those relations. This flexibility allows the annotation environment to be adapted to different research needs.

4 Annotation Workflow

In this section, we overview the annotation workflow using the proposed annotation environment.

4.1 Input data

The annotation process begins with loading the input data into the environment. The environment does not support span-based mention annotation, so the input data must be pre-annotated with spans for entity and event mentions. This can be done manually or with existing NLP applications. Once the input data is pre-annotated, it can be loaded into the environment and the annotation process can begin.

4.2 Event relation annotation

First step in the annotation is to reorder event mentions based on their temporal order. From the pre-annotated list of event mentions, annotators are shown a pair of events and their surrounding text, and asked to label pairwise relation by selecting a label among

- `independent`: no temporal relation between the event pair
- `before/after`: one event must be finished before the other starts
- `beginning/begun_by`: the beginning of one event is conditioned on the end of the other (e.g., *do X immediately after Y*)
- `ending/ended_by`: the end of one event is conditioned on the end of the other (e.g., *do X until Y*)
- `light-verb-construction (LVC)`[2]: not a temporal relation, but a lexical pattern that has two separate text parts

These temporal relation names (except for `LVC`, since `LVC` is not a temporal one) are selected from TimeML's `TLINK`, but the logic is largely based on Allen, 1983. Each pairwise annotation is then used in a simplified temporal reasoning algebra to generate next prompt in real-time[3].

4.3 Coreference link annotation

For this step, we directly adopt the previous CUTLER environment. Unlike other coreference annotation tools that require annotators to link entity mentions across the whole document, CUTLER decomposes the task to individual event-level and simplifies the complex conference task into an event-argument linking task. This means that annotators only need to link entity mentions that are part of the *current* event, which is much easier than linking mentions that are spread out across a document. We add improvements

[2]for `LVC` (e.g., *[**Bring**]*$_{evt}$ *to a [**boil**]*$_{evt}$), annotators are asked to pick the *head* event span.

[3]the algebra is only designed to reduce the number of pairwise prompts to annotators based on transitive reduction, and thus does not aim to construct a total order of events nor a full closure of temporal relations.

1. to event argument candidate selection algorithm to handle temporally conditioned overlapping events: previously there were no overlapping events, but in the new scheme, we have end temporal relations that indicate event overlaps.

2. to the real-time graph visualization feature, based on our addition of a temporal ordering annotation step to reflect the additional time dimension: fig 1 shows examples of independent and ending relations.

4.4 Coreference sub-type annotation

Although we proposed four sub-types of identity-based coreference previously, CUTLER only implements an explicit interface for annotating meronymy relations. Other sub-types are automatically inferred by some *magic* features of the tool, based on the entity types and event argument structures. We decided to re-do the coreference labeling interface to make the environment more flexible to different definitions of coreference types, as we found different label sets from different previous work. As a result, our environment asks annotators to explicitly pick a label when a coreference link is drawn, while keeping the original magic inference feature to provide some reasonable default values to annotators.

4.5 Output data

Annotation results are stored in relation triples, readily available for graph visualization for human readers or algorithmic ingestion for machine consumption.

5 Future work

At the moment, the environment is implemented as a locally hosted web application. However, we believe that our simplified annotation scheme and annotation workflow implementation will enable a quick adoption of the environment into crowd-source annotation tool running on platforms like AMT.

6 Conclusion

In this paper, we present an integrated annotation environment that supports different aspects of event semantics and coreference relations, including full- and near-identity sub-type labeling. The environment provides simplicity for annotators for quick and easy task completion, while provides flexibility for task designers who might need to adopt different typology and definitions of coreference relations for their research needs and interests. The scheme and environment (and related code) is publicly available as an open-source software. And our future direction is to port the theoretical concepts and tool interface to a more crowdsource-friendly implementation, to continue our effort to create an event-driven coreference annotation dataset.

Limitations

This work showcases our latest tool development efforts. The environment and annotation scheme we present in this work are highly tailored for modeling conferences and event semantics in procedural text, where the majority of events are transformational (that cause a changes in states of actual objects), rather than pragmatic or speculative, and the majority of entities have denotations to real-world objects. We are also developing a set of datasets that can be used to describe the semantics of these events and entities based on the environment and scheme described in this work. However, those dataset creation efforts are beyond the scope of this paper.

Ethics Statement

This work is based on our own previous work. The previous work included human annotation effort on publicly available dataset. The source data that we used in the annotation was collections of multicultural recipes text, written in English, distributed under Creative Common license. Given the data domain, annotation methodology, and tool development, we do not anticipate any major ethical concern.

Acknowledgements

We appreciate all the valuable feedback from the anonymous reviewers. This work was supported by NSF Award #2326985.

References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843.

Rahul Aralikatte and Anders Søgaard. 2020. Model-based annotation of coreference. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 74–79, Marseille, France. European Language Resources Association.

Jon Chamberlain, Massimo Poesio, and Udo Kruschwitz. 2016. Phrase detectives corpus 1.0 crowdsourced anaphoric coreference. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2039–2046, Portorož, Slovenia. European Language Resources Association (ELRA).

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.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

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, Dublin, Ireland. Association for Computational Linguistics.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Ankita Gupta, Marzena Karpinska, Wenlong Zhao, Kalpesh Krishna, Jack Merullo, Luke Yeh, Mohit Iyyer, and Brendan O'Connor. 2023. ezCoref: Towards unifying annotation guidelines for coreference resolution. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 312–330, Dubrovnik, Croatia. Association for Computational Linguistics.

Lynette Hirschman and Nancy Chinchor. 1998. Appendix F: MUC-7 coreference task definition (version 3.0). In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

Shinsuke Mori, Hirokuni Maeta, Yoko Yamakata, and Tetsuro Sasada. 2014. Flow graph corpus from recipe texts. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2370–2377, Reykjavik, Iceland. European Language Resources Association (ELRA).

Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In Sabine Braun, Kurt Kohn, and Joybrato Mukherjee, editors, *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pages 197–214. Peter Lang, Frankfurt a.M., Germany.

Bruno Oberle. 2018. SACR: A drag-and-drop based tool for coreference annotation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Massimo Poesio and Ron Artstein. 2008. Anaphoric Annotation in the ARRAU Corpus. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Massimo Poesio, Patrick Sturt, Ron Artstein, and Ruth Filik. 2006. Underspecification and Anaphora: Theoretical Issues and Preliminary Evidence. *Discourse Processes*, 42(2):157–175.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. 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.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27, Portland, Oregon, USA. Association for Computational Linguistics.

Jakob Prange, Nathan Schneider, and Omri Abend. 2019. Semantically constrained multilayer annotation: The case of coreference. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 164–176, Florence, Italy. Association for Computational Linguistics.

Marta Recasens, Eduard Hovy, and M Antonia Martı. 2010. A Typology of Near-Identity Relations for Coreference (NIDENT). In *LREC2010*.

Nils Reiter. 2018. CorefAnnotator - A New Annotation Tool for Entity References. In *Abstracts of EADH: Data in the Digital Humanities*.

Kyeongmin Rim, Jingxuan Tu, Bingyang Ye, Marc Verhagen, Eben Holderness, and James Pustejovsky.

2023. The Coreference under Transformation Labeling Dataset: Entity Tracking in Procedural Texts Using Event Models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12448–12460, Toronto, Canada. Association for Computational Linguistics.

Ina Roesiger, 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, Santa Fe, New Mexico, USA. 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.

Olga Uryupina, Ron Artstein, Antonella Bristot, Federica Cavicchio, Kepa Rodriguez, and Massimo Poesio. 2016. ARRAU: Linguistically-motivated annotation of anaphoric descriptions. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2058–2062, Portorož, Slovenia. European Language Resources Association (ELRA).

Antoine Widlöcher and Yann Mathet. 2012. The glozz platform: A corpus annotation and mining tool. In *Proceedings of the 2012 ACM Symposium on Document Engineering*, DocEng '12, page 171–180, New York, NY, USA. Association for Computing Machinery.

Xueqing Wu, Sha Li, and Heng Ji. 2023. OpenPI-C: A better benchmark and stronger baseline for open-vocabulary state tracking. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7213–7222, Toronto, Canada. Association for Computational Linguistics.

Amir Zeldes. 2022. Opinion Piece: Can we Fix the Scope for Coreference?: Problems and Solutions for Benchmarks beyond OntoNotes. *Dialogue & Discourse*, 13(1):41–62.