Linear Cross-document Event Coreference Resolution with X-AMR

Shafiuddin Rehan Ahmed, George Arthur Baker, Evi Judge, Michael Regan[†]
Kristin Wright-Bettner, Martha Palmer, and **James H. Martin**
University of Colorado, Boulder, CO, USA [†]University of Washington, Seattle, WA, USA
{shah7567, george.baker}@colorado.edu

Abstract

Event Coreference Resolution (ECR) as a pairwise mention classification task is expensive both for automated systems and manual annotations. The task's quadratic difficulty is exacerbated when using Large Language Models (LLMs), making prompt engineering for ECR prohibitively costly. In this work, we propose a graphical representation of events, X-AMR, anchored around individual mentions using a **cross**-document version of **A**bstract **M**eaning **R**epresentation. We then linearize the ECR with a novel multi-hop coreference algorithm over the event graphs. The event graphs simplify ECR, making it a) LLM cost-effective, b) compositional and interpretable, and c) easily annotated. For a fair assessment, we first enrich an existing ECR benchmark dataset with these event graphs using an annotator-friendly tool we introduce. Then, we employ GPT-4, the newest LLM by OpenAI, for these annotations. Finally, using the ECR algorithm, we assess GPT-4 against humans and analyze its limitations. Through this research, we aim to advance the state-of-the-art for efficient ECR and shed light on the potential shortcomings of current LLMs at this task. Code and annotations: https://github.com/ahmeshaf/gpt_coref

Keywords: semantics, discourse, events, coreference, model-in-the-loop annotation

1. Introduction

Event Coreference Resolution (ECR) involves identifying events that refer to the same real-world occurrence both within and across documents. Traditionally, ECR is performed on pairs of event mentions in a corpus through the use of rules, features, or neural methods to generate similarity scores (Kenyon-Dean et al., 2018), with neural methods such as Transformer-based encoders (Devlin et al., 2019; Liu et al., 2019; Beltagy et al., 2020) achieving state-of-the-art performance on various ECR benchmarks (Caciularu et al., 2021; Held et al., 2021). However, the quadratic nature of pairwise approaches makes it challenging to scale up to large corpora of thousands of documents.

Figure 1 presents three event mentions (m_1, m_2, and m_3) with their respective event triggers highlighted. m_1 and m_2 are examples of coreferent events, while m_3 is a related yet non-coreferent event. While m_1 and m_3 contain sufficient information required to make a negative coreferencing decision between them, additional extrasentential context is needed to determine the coreferential relationship between m_2 and the other two mentions.

The challenge of ECR stems from the inherent issue of establishing *singular terms* for event mentions (a, b) that can be compared for identity (*is a = b?*; Davidson (1969)). Consequently, pairwise methods resort to approximations of the coreference relationship between each mention pair by leveraging either the sentence or the entire document for contextual information. The methods that need to rely on the entire document for each pairwise decision (as in the case of m_2) are intractable on large corpora. We propose that by extracting the key semantics of mentions and by introducing a graphical structure between each mention, we can compress the information. This way we not only are able to create identifiers for the mentions that can be compared for sameness, but also make ECR completely linear in complexity.

Target Mention (m_1)

HP today announced that it has signed a definitive agreement to **acquire** EVT EYP Mission Critical Facilities Inc.

Target Mention (m_2)

Financial details of the **acquisition** EVT were not disclosed.

Target Mention (m_3)

Earlier this month Hewlett-Packard unveiled a bid of nearly $14 billion bid to **purchase** EVT Electronic Data Systems.

Figure 1: m_1 and m_2 are examples of coreferent mentions. m_3 although related to m_1 and m_2, is a different acquisition event.

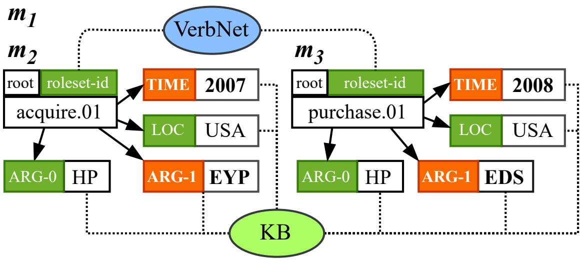


Figure 2: Compressed event semantics graphs of m_1, m_2, and m_3. The graph for m_2 is generated using the entire document in which it appeared. VerbNet classes are used for synonymous predicates. KB is used for argument coreference resolution.

Figure 2 illustrates the graph structure based on our new Cross-document Abstract Meaning Representations (X-AMR), inspired by Abstract Meaning Representation (AMR; Banarescu et al. (2013)). X-AMR captures event triggers and arguments linked using VerbNet Lexicon (Schuler, 2005) and a Knowledge Base (KB), resulting in corpus-level event graphs that we use to symbolically perform ECR without the need for pairwise scoring. Our specific contributions in this paper include:

- X-AMR annotations using the annotation tool provided by Ahmed et al. (2024) on the ECB+ corpus (Cybulska and Vossen, 2014).
- A novel ECR algorithm over the X-AMR graphs that avoids the computational cost of traditional pairwise approaches.
- An evaluation of the algorithm using gold-standard annotations for the ECB+ corpus.
- And finally, an evaluation of the approach with automatically generated X-AMR graphs using GPT-4 in zero-shot and few-shot settings with prompts based on a condensed version of the annotation guidelines of X-AMR.

Together, our annotations and findings suggest a promising path toward extending robust ECR to real-world applications where exhaustive pairwise approaches are not feasible.

2. Annotation Guidelines for X-AMR

We aim to annotate key event semantics with four arguments, ARG-0, ARG-1, ARG-Loc, and ARG-Time, capturing agent, patient (and theme), location, and temporal information. The selection of these arguments is to circumscribe an event by its *minimal participants* (Lombard, 2019; Guarino et al., 2022). We use the guidelines presented in the next section to hand annotate the roleset and argument information for the ECB+ train, development, and test sets using the standardized split of Cybulska and Vossen (2014). Following the annotation guidelines, we provide the enriched annotations of the ECB+ corpus by two Linguistic students. We use the prodi.gy-based X-AMR annotation tool provided by Ahmed et al. (2024)[1].

2.1. PropBank & AMR

Semantic role labeling (SRL) centers on the task of assigning the same semantic role to an argument across various syntactic constructions. For example, *the window* can be the (prototypical) Patient, or thing broken, whether expressed as syntactic object (*The storm broke the window*) or syntactic subject (*The window broke in the storm*).

```
agree.01 - agree          agree.01
   ARG-0: Agreer             ARG-0: HP
   ARG-1: Proposition        ARG-1: acquire.01
   ARG-2: Other entity          ARG-0: HP
               agreeing         ARG-1: EYP
```

Figure 3: The PropBank roleset definitions of agree.01 and the expected annotations in X-AMR.

The Proposition Bank (PropBank; Palmer et al. (2005); Pradhan et al. (2022)) has over 11,000 Frame Files providing valency information (arguments and their descriptions) for fine-grained senses of English verbs, eventive nouns, and adjectives. Figure 3 gives an example Frame File for *agree* as well as an instantiated frame for *HP has an agreement to acquire EYP*.

The resulting nested predicate-argument structures from PropBank style-SRL also form the backbones of AMRs, which in addition includes Named Entity (NE) tags and Wikipedia links (for 'HP' and 'EYP' in our example). AMRs also include explicit variables for each entity and event, consistent with Neo-Davidsonian event semantics, as well as inter- and intra-sentential coreference links to form directed, (largely) acyclic graphs that represent the meaning of an utterance or set of utterances.

Our enhanced X-AMR representation follows AMR closely with respect to NE and coreference, but stops short of AMR's additional structuring of noun phrase modifiers (especially with respect to dates, quantities and organizational relations), the discourse connectives and the partial treatment of negation and modality. However, we go further than AMR by allowing for cross-document coreference as well as multi-sentence coreference. X-AMR thus provides us with a flexible and expressive event representation with much broader coverage than standard event annotation datasets such as ACE[2] or Maven (Wang et al., 2020).

2.2. Roleset Sense Annotation

The first step in the annotation process involves identifying the roleset sense for the target event trigger in the given text. Annotators, using an embedded PropBank website and the assistance of the tool's model, select the most appropriate sense by comparing senses across frame files.
Handling Triggers with No Suitable Roleset: If there is no appropriate roleset that specifies the event trigger, particularly in cases when the trigger is a pronoun (it) or proper noun (e.g., Academy Awards), the annotator must then search for a roleset that defines the appropriate predicate.

[1] Readers are encouraged to check the original paper for details about the annotation tool

[2] https://www.ldc.upenn.edu/collaborations/past-projects/ace

2.3. Document-level Arguments Identification

Next, we identify the document and corpus-level ARG-0 and ARG-1 of the selected roleset. Annotators use the embedded PropBank website as a reference for the roleset's definition, ensuring that the ARG-0 (usually the agent) and ARG-1 (typically the patient) are consistent with the roleset's constraints. For arguments that cannot be inferred, the annotators leave those fields empty.

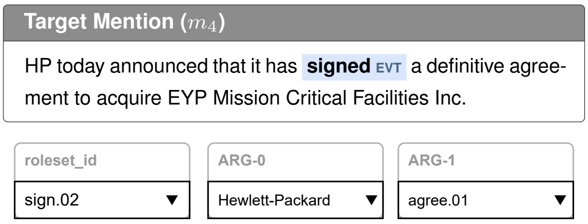
Within- and Cross-Document Entity Coreference Annotation: Annotators perform within- and cross-document entity coreference using a drop-down box of argument suggestions (suggested by the model-in-the-loop), simplifying coreference link establishment. In difficult cases like m_2 (Fig 1), where ARG-0 and ARG-1 are missing, the drop-down box helps by suggesting “HP" and “EYP" from the m_1 sentence. Similarly, in m_4 (Figure 4), the drop-down box assists in resolving ARG-0 (it) as “HP", using the information earlier within the sentence. Annotators are also allowed to input multiple values separated by “/" as needed, (e.g., if two people performed some action together, "Person 1/Person 2").

Nested ARG-1: In many cases, the ARG-1 may itself be an event. In such cases, the annotator is tasked with identifying the head predicate of the ARG-1 role and providing its corresponding roleset ID. We then search for the annotations for such an ARG-1 and connect it to the target event. Fig 4 has an example of a mention with an eventive ARG-1. For this, the annotator needs to provide the roleset for the predicate of the ARG-1 clause (agree.01) as the ARG-1 in this annotation process.

ARG-Loc & ARG-Time Identification Annotators may also utilize external resources, such as Wikipedia[3], or Google-News, for the accurate identification of temporal and spatial arguments. This is required when the document does not explicitly mention the location and time of the event.

3. Human Annotations

To perform the X-AMR annotations, we employ two annotators, and we execute this process in a systematic two-step approach. In the initial phase, these annotators are responsible for identifying the roleset ID associated with each event trigger. We aggregate all event mentions for which both annotators have concurred on the same roleset ID. For those instances where there is a lack of consensus between the annotators, we enlist the assistance of an adjudicator to resolve the discrepancies. The annotations that have been finalized, either through agreement or adjudication, are then collectively advanced to the subsequent task of identifying the arguments.



Target Mention (m_4)

HP today announced that it has **signed** EVT a definitive agreement to acquire EYP Mission Critical Facilities Inc.

roleset_id	ARG-0	ARG-1
sign.02 ▼	Hewlett-Packard ▼	agree.01 ▼

Figure 4: Eventive ARG-1 in m_4 for the roleset sign.02. The ARG-1 clause is annotated as the connecting event with roleset ID agree.01

3.1. Annotation Analysis

We have currently annotated all the mentions in the corpus with their Roleset IDs and 5,287 out of the 6,833 with X-AMR. In the three splits, only the Dev set has been fully annotated. We calculate the inter-annotator agreement (IAA) on the common Roleset predictions. The IAA is highest for the Dev set at 0.91, as depicted in Table 1. Consequently, we utilize the Dev set as our benchmark for experiments in the following sections.

	Train	Dev	Test
Documents	594	196	206
Mentions	3808	1245	1780
Roleset ID Agreement	0.84	**0.91**	0.80
w/ X-AMR	3195*	1245	847*
w/ Nested ARG-1	1081	325	220
w/ ARG-Loc	2949	1243	707
w/ ARG-Time	3192	1244	805

Table 1: Corpus statistics for event mentions in ECB+ and the mentions annotated with X-AMR (*Annotation in Progress). Inter-annotator agreement for the Roleset ID is highest for the Dev set.

Arguments: Our analysis reveals a significant presence of mentions with nested ARG-1 annotations, as highlighted in Table 1 (w/ Nested ARG-1). This underscores the importance of capturing nested event relationships effectively. Additionally, our annotations for location and time modifiers successfully capture this information for nearly all mentions (w/ X-AMR), thanks to the assistance provided by drop-down options and the model-in-the-loop approach. These tools are particularly valuable in cases where date references are not explicitly mentioned in the document.

[3] Although we add this in the guidelines, the annotators do not wikify. This is only for GPT to generate instructions for itself. Our choice is to use Wikipedia over the more commonly used KB-wikidata because of GPT-friendly identifiers of the pages.

4. Graph-based ECR Algorithm

Our proposed approach for ECR builds upon previous research efforts that use minimum participants. Cybulska and Vossen (2013) utilize heuristics to ascertain event relationships based on various factors, such as location, time, and participant compatibility. Choubey and Huang (2017) employ iterative techniques to identify event relations, both within and across sentences. It's important to note that both of these approaches are pairwise methods and do not incorporate cross-document entity coreference into their methodologies. In contrast, our approach with X-AMR not only leverages cross-document entity coreference but also capitalizes on AMR's nested event structure for ECR.

4.1. EID: Event Identifiers

We generate EID using the roleset, ARG-0, and ARG-1. To evaluate the influence of location and time, we produce $\mathtt{EID}_{\mathrm{lt}}$ by incorporating ARG-Loc and ARG-Time. These identifiers facilitate comparison between two events, allowing coreference resolution by matching the identifiers. Specifically, two events (m_i, m_j) are deemed coreferent (where $\mathtt{coref}(m_i, m_j)$ is true) if any of their identifiers match in $\mathtt{EID}(m_i)$ and $\mathtt{EID}(m_j)$ match, as illustrated in Equation 1.

$$\mathtt{coref}(m_i, m_j) \equiv \mathtt{EID}(m_i) \cap \mathtt{EID}(m_j) \neq \varnothing \quad (1)$$

Even though Eq 1 is represented pairwise, we design the clustering algorithm by first creating buckets of mentions with the same identifiers. This way we generate a *sparse binary similarity matrix* of only the pairs of mentions in the same buckets representing the EIDs.

4.1.1. EID Generation

We generate the identifiers differently for Standard events (m, like m_1, m_2, m_3) and Nested Events (m_e, such as m_4 and m_5). For standard events, the identifier ($\mathtt{EID}_0$) is constructed by merging the ARG-0, roleset ID (PB), and ARG-1 as shown in Equation 2. For instance, $\mathtt{EID}(m_1)$ is denoted as ⟨HP, acquire.01, EYP⟩.

$$\mathtt{EID}_0(m) = \langle\ \text{ARG-0}(m),\ \mathtt{PB}(m),\ \text{ARG-1}(m)\rangle \quad (2)$$

In the case of Nested Events (ARG-1 is also an event), we employ a recursive strategy to generate identifiers. Specifically, we produce multiple EIDs by traversing the arguments of nested events up to a maximum depth, N, as delineated in Equations 3 and 4. This method aims to connect the root event to a standard ARG-1 within the event chain. This procedure is denoted as $\mathtt{EID}_n$, where n indicates the utilized depth. Notably, for our experiments, we set $n = \mathrm{N}$ during EID generation.

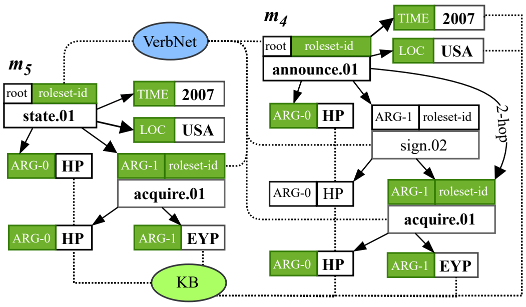


Figure 5: Event Identifier and Coreference for events with eventive ARG-1. The $\mathtt{EID}_2(m_4)$ is equivalent $\mathtt{EID}_1(m_5)$ with the help of VerbNet to detect synonymy and KB to link arguments.

$$\begin{aligned}\mathtt{EID}_n(m_e) = \bigcup_{k}^{n} \Big[& \langle \text{ARG-0}(m_e), \mathtt{PB}(m_e)\rangle \times \\ & \mathtt{EID}^{\text{hop}}_{k,n}(\text{ARG-1}(m_e)) \Big],\end{aligned} \quad (3)$$

where × is the Cartesian product for generating all the concatenations of the tuples.

$$\mathtt{EID}^{\text{hop}}_{k,n}(m) = \begin{cases} \text{ARG-1}(m) & \text{if standard,} \\ & \text{or } k = n, \\ \mathtt{EID}_{n-1}(\text{ARG-1}(m)) & \text{otherwise} \end{cases} \quad (4)$$

Using Eq 3 and 4, we generate the identifiers for m_4 and m_5 (Figure 5) as shown below:

$\mathtt{EID}_2(m_4)$	⟨HP, announce.01, HP, sign.02, HP, acquire.01, EYP⟩, ⟨HP, announce.01, HP, acquire.01, EYP ⟩
$\mathtt{EID}_1(m_5)$	⟨HP, state.01, HP, acquire.01, EYP ⟩

The 2-hop identifier of m_4 is exactly same as the 1-hop one of m_5, except for the roleset IDs. To detect synonymy between the rolesets, we use VerbNet ($\mathtt{PB}_{\mathtt{HVN}}$), and we maintain a KB to link the arguments. With the combination of all these components, we can infer that m_4 is coreferent with m_5.

We also use the ARG-Loc and ARG-Time to separately generate an identifier, $\mathtt{EID}_{\mathrm{lt}}$, for both kinds of events as shown in Equation 5.

$$\begin{aligned}\mathtt{EID}_{\mathrm{lt}}(m) = \ & \langle \text{ARG-0}(m),\ \mathtt{PB}(m), \\ & \text{ARG-Loc}(m),\ \text{ARG-Time}(m)\rangle\end{aligned} \quad (5)$$

4.2. Clustering Methods

We generate the adjacency matrix of the mentions by using certain baselines and the event identifiers. The adjacency matrix is then used for hard-clustering the events by finding the connected components.

Baseline (LEM): Clustering mentions with the same lemma for their triggers serves as our baseline method.

Rolesets IDs (PB$_{\text{H}}$, PB$_{\text{G}}$): We cluster mentions based on the strict similarity of these PropBank Roleset IDs. We use the human (PB$_{\text{H}}$) and GPT-generated (PB$_{\text{G}}$, see §6) roleset IDs separately.

RS with VerbNet Syn classes (PB$_{\text{HVN}}$, PB$_{\text{GVN}}$): We cluster mentions of synonymous rolesets based on VerbNet Classes (Brown et al., 2011, 2022) allowing less strict roleset matching.

Event Identifiers: We vary the EID methods in the following ways:

EID$_{\text{N}}$: We cluster with EID$_{\text{N}}$ using PB$_{\text{H}}$ or PB$_{\text{G}}$

EID$_{\text{lt}}$: We cluster with EID$_{\text{lt}}$ using PB$_{\text{H}}$ or PB$_{\text{G}}$

EID$_{\text{N}}^{\text{VN}}$: EID$_{\text{N}}$ only, but with PB$_{\text{HVN}}$ or PB$_{\text{GVN}}$ to identify roleset classes for PB$_{\text{H}}$ or PB$_{\text{G}}$

EID$_{\text{lt}}^{\text{VN}}$: EID$_{\text{lt}}$ only, but with PB$_{\text{HVN}}$ or PB$_{\text{GVN}}$ to identify roleset classes for PB$_{\text{H}}$ or PB$_{\text{G}}$

EID$_{\text{N}}$ $\wedge$ EID$_{\text{lt}}$: We cluster the mentions when they have the same EID$_{\text{N}}$ and EID$_{\text{lt}}$.

EID$_{\text{N}}$ $\vee$ EID$_{\text{lt}}$: We cluster the mentions when they have either the same EID$_{\text{N}}$ or EID$_{\text{lt}}$.

We also include the VerbNet class versions of the final two methods.

In addition to the individual methods listed above, we also employ combinations of methods on the annotations of the two annotators (A$_1$, A$_2$). $\wedge$-clustering employs the rule that two mentions should have the same annotations from A$_1$ and A$_2$ (A$_1 \wedge$ A$_2$). $\vee$-clustering employs the rule that any of the two annotators's annotations could be the same for two mentions (A$_1 \vee$ A$_2$).

5. ECR Results of A$_1$ and A$_2$

We use the standard clustering metrics for ECR (MUC, B^3, $CEAF_e$, and CoNLL F1—the average of MUC, B^3 and $CEAF_e$; Vilain et al. (1995); Bagga and Baldwin (1998); Luo (2005); Denis and Baldridge (2009); Luo et al. (2014); Pradhan et al. (2014); Moosavi et al. (2019)). To evaluate recall, we compute the mean recall values from MUC and B^3 (R$_{avg}$). Similarly, our precision metric, P$_{avg}$ is derived from the average precision values of MUC and B^3. Our primary measure of overall performance is CoNLL F1. We applied various algorithmic methods to the ECB+ development set, which has been annotated using the X-AMR framework by A$_1$ and A$_2$. Each annotator's performance is independently assessed, along with the $\vee$-clustering approach, A$_1 \vee$ A$_2$. We collate the results in Table 2[4].

	Method	R$_{avg}$	P$_{avg}$	CoNLL
	LEM	72.6	64.0	63.7
	PB$_{\text{H}}$	81.2	63.5	66.1
	PB$_{\text{HVN}}$	**91.0**	43.0	44.9
A$_1$	EID$_{\text{N}}$	75.6	90.5	78.4
	EID$_{\text{lt}}$	77.9	91.2	**79.8**
	EID$_{\text{N}}$ $\wedge$ EID$_{\text{lt}}$	74.2	**92.8**	78.4
	EID$_{\text{N}}$ $\vee$ EID$_{\text{lt}}$	79.3	88.9	**79.8**
	EID$_{\text{N}}^{\text{VN}}$ $\wedge$ EID$_{\text{lt}}^{\text{VN}}$	80.0	84.2	78.5
A$_2$	EID$_{\text{N}}$	69.8	89.4	75.0
	EID$_{\text{lt}}$	66.6	88.8	72.1
	EID$_{\text{N}}$ $\wedge$ EID$_{\text{lt}}$	61.0	91.8	69.7
	EID$_{\text{N}}$ $\vee$ EID$_{\text{lt}}$	76.0	86.4	77.1
A$_1 \vee$ A$_2$	EID$_{\text{N}}$	79.0	82.9	77.4
	EID$_{\text{lt}}$	80.2	83.6	77.8
	EID$_{\text{N}}$ $\wedge$ EID$_{\text{lt}}$	78.4	85.6	78.3
	EID$_{\text{N}}$ $\vee$ EID$_{\text{lt}}$	80.8	80.8	76.9
	EID$_{\text{N}}^{\text{VN}}$ $\wedge$ EID$_{\text{lt}}^{\text{VN}}$	**86.9**	69.1	73.1

Table 2: ECR results comparing the annotators on the Development Set of the ECB+ Corpus. We report the baseline results using only the lexical information, and, the ECR performance of the proposed graph-based algorithm on the X-AMR annotations of A$_1$ and A$_2$, and, a union of A$_1$ and A$_2$ (A$_1 \vee$A$_2$). Boldened are the interesting results.

From the table, it is evident that utilizing the roleset IDs (PB$_{\text{H}}$) achieves a better result than lemmas (LEM). Even though PB$_{\text{HVN}}$ has the highest recall of 91%, the overall performance is quite low. The 9% recall error indicates the gap in the VerbNet class annotations for all the PropBank rolesets. This suggests there may be room for refining the VerbNet-Pro annotations for better compatibility (Spaulding et al., 2024).

Comparing annotators, A$_1$ provided more accurate annotations than A$_2$, particularly in identifying location and time elements. Both annotators performed best with the EID$_{\text{N}}$ $\vee$ EID$_{\text{lt}}$ setting, with A$_1$ recording the best CoNLL F1 score of 79.8%. A$_2$'s annotations would need further refinement in order to match A$_1$'s recall. When considering precision, the EID$_{\text{N}}$ $\wedge$ EID$_{\text{lt}}$ method stood out, with A$_1$registering the highest precision at 92.8%.

For A$_1 \vee$ A$_2$, the results are mixed. Although the recall is consistently higher than any individual annotator, it does not beat A$_1$'s best CoNLL. This method achieves the best recall of 86.9 when used in conjunction with the VerbNet classes while also

[4] A$_1 \wedge$ A$_2$ results are excluded due to inferior quality.

having a CoNLL F1 greater than 70%. The mixed results for the combined method underscore the complexities involved in integrating and harmonizing annotations from different sources.

A CoNLL F1 of 80% seems to be an upper bound for a purely symbolic approach for ECR. However, we want to stress that after annotating X-AMR, we are in effect collecting free ECR annotations (eg. 75% coreference links with 93% precision). ECR annotations are traditionally done in a pairwise manner, an approach that is tedious and error-prone (Song et al., 2018; Wright-Bettner et al., 2019). In contrast, X-AMR has an annotator-friendly methodology where an annotator would need to read a particular event mention typically only once. It also avoids annotation errors cascading into subsequent mentions as demonstrated by the high precision of our method.

6. GPT-4 as Annotator

Recent work in prompt engineering converts a text-based natural language task to a corresponding structured prediction task. In this spirit, we create prompts for extracting the X-AMR graph for a specific event, by providing the instructions for the task, exemplars for the structure of the response, and the right context for in-context zero-shot learning. Due to budget and time constraints, we make a smaller subset (120 mentions) of the development set (dev$_{\text{small}}$) to run our experiments on. We then assess the performance of GPT-4 (September 27, 2023 version) against the human annotations for this subset. We try two prompt engineering techniques (G$_1$, G$_2$) to extract the X-AMR graphs of the events, and use the `EID` generation and clustering methods from §4.2.

6.1. G$_1$: Prompt Engineering

For G$_1$, we use a straightforward approach to generate the prompts. We start by generating a list of instructions. As shown in Figure 7, we arrive at five instructions by condensing the relevant sections of the annotation guidelines. We adopt a semi-automated way of generating the instructions, in which we first pass the relevant sections to ChatGPT and then hand-correct its output.

6.1.1. Structured Prediction: Label Definitions

We then prompt GPT to produce a JSON output as the response. We offer detailed definitions for the keys in the JSON string, as illustrated in Figure 6. Additionally, we incorporate the coreference key and prompt GPT to generate Wikipedia links in the format “/wiki/Title_Name". Labels for Chain of Thought reasoning (Wei et al., 2022) are also included, addressing questions like “Is it a Nested Event?", “What is the event trigger?", “Who are the participants?", and “When and where did the event take place?". The final key in the list is “Event Description" that is a way to prompt GPT to produce a concise sentence encapsulating the event arguments including Time and Location.

Label Definitions

Here are the definitions of the keys in the JSON output:

Roleset ID: The PropBank Roleset ID corresponding to the event trigger

ARG-0: The text in the Document corresponding to the typical agent

ARG-0 Coreference: The reference to the ARG-0 in Wikipedia in the format /wiki/Wikipedia_ID

:

ARG-1 Roleset ID: If the Event is Nested, provide the Roleset ID for the head event in ARG-1 clause

ARG-Location: The reference to the event location in Wikipedia

ARG-Time: The event time in the format of Month-Day-Year in your knowledge of the world or the document

Event Description: In a single sentence, summarize the event capturing the Roleset_ID and the names and wiki links of the Participants, Location and Time

Figure 6: Label definitions for the event's Roleset ID and the Arguments that include the Wikipedia links. Event Description is a single sentence encapsulating the key components of the event.

Annotation Instructions

You are a concise annotator that follows these instructions:

1. Identify the target event trigger lemma and its correct roleset sense in the given text.
2. Annotate the document-level ARG-0 and ARG-1 roles using the PropBank website for the roleset definitions.
3. If the ARG-1 role is an event, identify the head predicate and provide its roleset ID.
4. Perform within-document and cross-document anaphora resolution of the ARG-0 and ARG-1 using Wikipedia.
5. Use external resources, such as Wikipedia, to annotate ARG-Loc and ARG-Time.

Figure 7: The condensed annotation instructions serve as a guide for GPT-4 in its generation of X-AMR event extraction.

Finally, we add the entire document of the event and the sentence with the marked trigger (phrase in the sentence sorrounded by <m> and </m>) as context, and then prompt GPT to generate the corresponding JSON response.

6.2. G$_2$: Prompt Engineering

A challenge observed in G$_1$ is its inability to determine specific pieces of information, particularly, ‘Location’ and ‘Time’ when they are absent within the source document. To address this shortfall, we introduce a complementary method, G$_2$.

Event Descriptions: In G_2, instead of relying solely on the document's raw content, we incorporate additional context derived from the event descriptions of what we term as *complete* events. These complete events are identified across all documents related to a specific topic at the prediction stage. A *complete* event is characterized by having all its requisite arguments, including Time and Location, predicted by G_1.

De-duplication: To enhance the quality and relevancy of this list, any coreferent events (duplicates) are eliminated.

Event List in Context: With the refined list, we pivot from using the entire document as context (as practiced in G_1) to utilizing this labeled list of Event Descriptions. Furthermore, the description of the current target event is also included.

Best Matching Event Description: We introduce a label called "Best Matching Event Description" at the beginning of prediction. This label pinpoints the most comprehensive and relevant description in correlation to the target mention. The intention behind this is to direct GPT's attention to a singular event description, enabling it to supplement the arguments not identified by G_1.

In essence, G_2 furnishes a richer context, combining aggregated information from various documents, to rectify the limitations observed in G_1.

7. ECR Results of G_1 and G_2

We compare the methods G_1 and G_2 (The cost for running G_1 was \$4, and G_2 was \$6.) separately with A_1 (the annotations with better quality among the two annotators) on $\text{dev}_{\text{small}}$. As shown in Table 3, the roleset identification by GPT-4 is impressive, thereby we only see a 3 point difference between $\texttt{PB}_{\texttt{H}}$ and $\texttt{PB}_{\texttt{G}}$. In $\text{dev}_{\text{small}}$, we observe the results are bounded by recall, therefore we use the VerbNet class approaches.

When using $\texttt{EID}$ methods, A_1 achieves the best CoNLL F1 of 83.6. When it comes to GPT-4, both G_1 and G_2, fell terribly short of A_1 and do not even surpass the Roleset ID baseline ($\texttt{PB}_{\texttt{G}}$), with G_1's best performance is short by 25 points and G_2 by 15. The shortcoming of G_2 can mainly be attributed to the failure of capturing nested events (only 5 of the 26 nested event arguments were identified). Interestingly, these methods consistently improve performance over the VerbNet baseline ($\texttt{PB}_{\texttt{GVN}}$). Between G_1 and G_2, we see a large performance increase (G_2 over G_1 by 10 points), emphasizing the benefits of using corpus-level Event Descriptions in the prompts.

The results reveal the limitations of GPT-4 on this task. However, efficient usage of corpus-level information in generating X-AMR graphs lays out an exciting path forward for future work.

	Method	R_{avg}	P_{avg}	CoNLL
	$\texttt{LEM}$	57.2	84.8	65.1
	$\texttt{PB}_{\texttt{H}}$	72.2	85.7	**75.3**
	$\texttt{PB}_{\texttt{G}}$	70.6	80.6	72.4
	$\texttt{PB}_{\texttt{HVN}}$	**90.4**	51.8	55.9
	$\texttt{PB}_{\texttt{GVN}}$	87.6	46.7	49.3
A_1	$\texttt{EID}_{\text{N}}$	68.8	**100**	77.7
	$\texttt{EID}_{\text{N}}^{\texttt{VN}}$	78.4	97.2	**83.6**
	$\texttt{EID}_{\text{N}}^{\texttt{VN}} \vee \texttt{EID}_{\texttt{lt}}^{\texttt{VN}}$	80.8	93.3	83.4
G_1	$\texttt{EID}_{\text{N}}^{\texttt{VN}}$	41.8	88.9	53.6
	$\texttt{EID}_{\texttt{lt}}^{\texttt{VN}}$	37.2	83.0	49.4
	$\texttt{EID}_{\text{N}}^{\texttt{VN}} \vee \texttt{EID}_{\texttt{lt}}^{\texttt{VN}}$	51.2	83.8	58.0
G_2	$\texttt{EID}_{\text{N}}^{\texttt{VN}}$	51.8	90.6	62.2
	$\texttt{EID}_{\texttt{lt}}^{\texttt{VN}}$	57.0	87.4	65.3
	$\texttt{EID}_{\text{N}}^{\texttt{VN}} \vee \texttt{EID}_{\texttt{lt}}^{\texttt{VN}}$	63.4	86.1	**68.4**

Table 3: ECR results comparing A_1 with the two prompting methods G_1 and G_2, on $\text{dev}_{\text{small}}$. We report the baseline results using only the lexical information, and, the ECR performance after leveraging X-AMR. Boldened are the interesting results.

8. Analysis

8.1. Algorithm Complexity

We conduct an artificial experiment to empirically demonstrate the ECR algorithm's linear complexity when coupled with X-AMR. In this experiment, we expand the annotated event mentions of the development set (by duplicating) to create a sizable collection comprising 200,000 mentions. Next, we systematically execute the algorithm across varying ranges, from 60,000 to 200,000 mentions. For each iteration, we measure the time the ECR algorithm

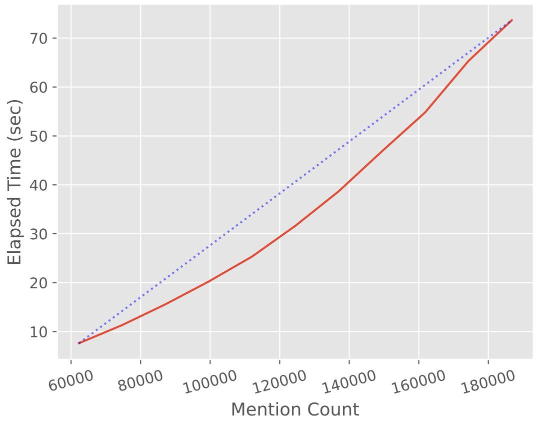


Figure 8: X-AMR ECR Algorithm running time on synthetic mentions. The dotted line is used as a reference to check the linearity of the algorithm (red line).The complexity is roughly linear.

Error Category	% Error	Snippet and Explanation
Annotation Error	1.6	m_1: The man is thought to have **fallen** much earlier in the day. m_2: [...] Duncan Rait died after slipping and **falling** [...] Explanation: annotator misidentifies"the man" as "Duncan Rait".
ECB+ Annotation Error	5.8	m_1: [...] the Philly Sixers **canned** Jim O'Brien [...] m_2: Jim O'Brien was **terminated** from [...] Ohio State University [...] Explanation: In ECB+ these events are falsely labelled as coreferent.
Incorrect VerbNet Class	9.1	[...] who was **gunned down** at their office Christmas party. Explanation: new Propbank rolesets aren't yet mapped to VerbNet.

Table 4: Qualitative Error Analysis of the ECR Algorithm for A_1 on $\text{dev}_{\text{small}}$. In Snippet, the event triggers are in **bold font**, and the key texts that help recognize the errors are underlined.

takes to run and depict the plot in Figure 8. The figure shows that the algorithm's running time is roughly linear. Efficiency-wise, the algorithm would take under 30 seconds even when the number of mentions surpasses 100,000, thus presenting a tractable solution for ECR at scale.

8.2. Error Analysis

We analyze the errors made by our system by examining the clustering decisions based on the A_1, G_1, and G_2 annotations of $\text{dev}_{\text{small}}$ (121 mentions).

In the human-annotated A_1, we observe that ∼1.6% (2/121) of the misclassifications were due to annotator misinterpretations of the passages. In another ∼5.8% (7/121) of cases, an incorrect cluster is assigned due to errors in the original ECB+ dataset's labels, which are made evident by mismatched X-AMR arguments. For example, the ECB+ labels erroneously consider football coach Jim O'Brien's separate terminations from Ohio State University and the Philadelphia 76ers as the same event. Finally, ∼9.1% (11/121) are misclustered due to PropBank labels which do not yet exist in VerbNet and so do not belong to a class; e.g. the newer "opening fire" roleset wasn't identified as being of the same class as "shoot". Examples of each type of error are provided in Table 4.

In addition to the problems faced by human annotators, the machine-created annotations G_1 and G_2 suffered heavily from their inability to access external resources to resolve relative times, locations, and references to entities, in addition to inconsistent annotations (which the human annotators did not suffer from due to the saved argument drop-down), e.g ['South_Richmond_Hill,_Queens', 'Queens', 'Richmond_Hill,_Queens'] all refer to the same place[5].

9. Limitations & Future Work

One limitation of our approach is that we require the PropBank resource for a particular language. In addition, the annotation tool is for-pay software. However, PropBank now has annotations for Chinese, Arabic, Urdu, Hindi, French, German, Spanish, and Catalan, and the annotation tool also works on a variety of languages. Our future work involves annotating X-AMR on a Spanish corpus.

The annotation tool released by Ahmed et al. (2024) omits a lot of AMR information (e.g. modality and negation), sticking strictly to the concept of minimal information for ECR. We also do not empirically demonstrate the efficiency of the model-in-the-loop annotations in this work. We leave the tool enhancements, including the incorporation of GPT-in-the-loop and a thorough analysis of the annotation efficiency (like Cai et al. (2023) and Ahmed et al. (2023b)) for future work.

The results for both human and GPT-annotated approaches fall short of state-of-the-art techniques for ECR that involve heuristics (for filtering) and fine-tuning BERT in a pairwise manner (Held et al., 2021; Ahmed et al., 2023a). We hypothesize that the X-AMR annotations might be beneficial to the heuristic-based filtering step in these methods. The Event Description generated by G_1 could also be employed while fine-tuning BERT which we believe is an interesting direction for neuro-symbolic methods for ECR.

Two main issues of using GPT-4 in our work are Data contamination (Magar and Schwartz, 2022; Wu et al., 2023) and reproducibility. Since both PropBank and ECB+ are publicly available resources, it is most likely that the test sets might be part of its pre-training data. We argue that since our task is vastly different from the pretraining task, the effect of contamination is minute as demonstrated by the results. Reproducibility, however, is a much greater limitation. By providing the GPT-4 output

[5] For a more comprehensive list of examples, please refer to the provided Excel file in the repository

on the train set (will release this upon acceptance), we set a mechanism to distill the knowledge into smaller in-house reproducible models like LIAma (Touvron et al., 2023) (Or even much smaller traditional auto-regressive models like FLAN-T5 (Chung et al., 2022)) for future work.

Finally, we limit the scope of our work to gold mentions instead of predicted mentions (Cattan et al., 2021). As a result, we could not compare X-AMR directly with the output of standard AMR parsers (Flanigan et al., 2014). Future work can approach this in a two-step way, with the first step being trigger identification, and then we can employ X-AMR on the predicted mentions.

10. Related Work

Document-level event extraction and event extraction with prompts (Li et al., 2021; Yang et al., 2022a; Xu et al., 2022) has been a major source of inspiration for our work. We extend this methodology for a more comprehensive cross-document level extraction by taking into account the named and unnamed arguments from previously seen documents into the annotation framework.

The Generative Pre-trained Transformer (GPT; Radford et al. (2018, 2019)) is an auto-regressive Transformer (Vaswani et al., 2017) language model developed by OpenAI, demonstrating exceptional performance across various natural language processing tasks. It uses a unidirectional, self-attention mechanism for effective context representation and is pre-trained on extensive unsupervised text corpora. The model follows a two-stage process of pre-training and fine-tuning, allowing it to adapt to specific tasks with minimal labeled data. GPT has undergone several iterations, with GPT-4 (OpenAI, 2023) being the most recent.

In recent years, research has increasingly focused on evaluating GPT's performance in multitask and zero/few-shot learning scenarios (Brown et al., 2020; Kojima et al., 2023). For instance, the study conducted by Radford et al. (2019) assesses the effectiveness of various LLMs in a zero-shot learning setting. Their findings imply that these models have the potential to equal, if not exceed, the performance of existing baselines on a range of NLP benchmarks.

Our objective is to underscore the importance of X-AMR with a focus on event coreference resolution, which integrates PropBank (Palmer et al., 2005; Pradhan et al., 2022) SRL as an intermediate phase. This approach is motivated by GPT-4's capability to produce free-text SRL for individual events (Zhang et al., 2022) instead of directly generating interconnected event graphs, as necessitated by AMR. By leveraging GPT-4's strengths, our suggested method can offer a more thorough and effective representation of events in a given text while preserving their structure and relationships, and therefore facilitate ECR.

Besides the hybrid approach and prompt engineering, we also stress the need for a linear algorithm over a quadratic ECR method, utilizing the generated graphs. Quadratic ECR with GPT (i.e., binary coreference decision between mention pairs) has produced negative outcomes, as evidenced by Yang et al. (2022b). Furthermore, this method would be expensive, potentially costing hundreds of dollars to execute using GPT-4. By adopting a linear algorithm, we aim to address these limitations, offering a more cost-effective and efficient solution for ECR. We propose a linear graph-based method for ECR using the generated key semantic information for the event mentions.

Over time, efforts have been made to enrich event datasets, such as the Richer Event Descriptions (RED; O'Gorman et al. (2016)) corpus and the Event Coref Bank plus corpus (ECB+; Cybulska and Vossen (2014)). The RED corpus enhanced ERE (Song et al., 2015) annotations by marking coreference for entities, events, and times, as well as temporal, causal, and subevent relationships in partial coreference through a multi-stage pipeline. In ECB+, Cybulska and Vossen (2014) expanded event descriptions by adding event classes with specific entity types and times, as well as inter-/intra-document coreference, to better represent the events within the ECB corpus (Bejan and Harabagiu, 2010). In a similar light, we enrich the ECB+ corpus with the X-AMR annotations with the goal of making ECR efficient and as a way to assess the performance of GPT-4.

11. Conclusion

In this paper, we introduced X-AMR, a corpus-level version of AMR. We provided a new model-in-the-loop tool with which we enriched the ECB+ corpus with X-AMR annotations. We then introduced a novel linear graph-based ECR algorithm that leverages the nested event structure and the cross-document entity coreference of X-AMR. The annotations coupled with the algorithm serve as a way for linearly generating cross-document event coreference annotations, cutting through a very challenging task. Finally, we developed two prompt engineering approaches for GPT-4 to automatically produce X-AMR graphs. We then compared the results against human annotations and showed limitations of GPT-4 on this task. We also provide comprehensive and concise GPT-generated event descriptors in this process that we believe have a lot of utility in other event tasks. Collectively, our contributions pave a path toward efficient ECR methods and their corresponding annotations.

Ethics Statement

Recognizing the rigor and tediousness of the annotation process, our research ensured that all annotators were fairly compensated, given reasonable work hours, and provided with regular breaks to maintain consistency and quality. Comprehensive training and clear guidelines were offered, and a robust communication channel was established to address concerns, ambiguities, and to encourage feedback. Our team made efforts to involve a diverse group of annotators to minimize biases.

To alleviate the monotonous nature of the task, we employed user-friendly tools, rotated tasks, and supported peer discussions. We also acknowledged the crucial role of annotators in our research, ensuring their contributions were recognized and valued. Post-task, a summary of our findings was shared with the annotators, incorporating their feedback into the final manuscript, underlining our commitment to an inclusive and ethical research approach.

By adhering to the LREC-COLING guidelines, we aim to emphasize the ethical considerations surrounding the involvement of annotators in research projects. We believe that a humane, respectful, and inclusive approach to data annotation not only results in superior-quality datasets but also upholds the dignity and rights of all involved.

Acknowledgements

We want to thank the reviewers of LREC-COLING 2024 who helped improve this paper. Part of this work was done during an internship of one of the authors at ExplosionAI GmbH. We would also like to thank Ákos Kádár, Matthew Hannibal, Nikhil Krishnaswamy, Elizabeth Spaulding, and the BoulderNLP group for their valuable comments on this paper. We gratefully acknowledge the support of DARPA FA8750-18-2-0016-AIDA – RAMFIS: Representations of vectors and Abstract Meanings for Information Synthesis and a sub-award from RPI on DARPA KAIROS Program No. FA8750-19-2-1004. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or the U.S. government.

12. Bibliographical Resources

Shafiuddin Rehan Ahmed, Jon Cai, Martha Palmer, and James H. Martin. 2024. X-AMR annotation tool. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 177–186, St. Julians, Malta. Association for Computational Linguistics.

Shafiuddin Rehan Ahmed, Abhijnan Nath, James H. Martin, and Nikhil Krishnaswamy. 2023a. $2 * n$ is better than n^2: Decomposing event coreference resolution into two tractable problems. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1569–1583, Toronto, Canada. Association for Computational Linguistics.

Shafiuddin Rehan Ahmed, Abhijnan Nath, Michael Regan, Adam Pollins, Nikhil Krishnaswamy, and James H. Martin. 2023b. How good is the model in model-in-the-loop event coreference resolution annotation? In *Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII)*, pages 136–145, Toronto, Canada. Association for Computational Linguistics.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Cosmin Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422, Uppsala, Sweden. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

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.

Susan Windisch Brown, Dmitriy Dligach, and Martha Palmer. 2011. VerbNet class assignment as a WSD task. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew Peters, Arie Cattan, and Ido Dagan. 2021. CDLM: Cross-document language modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2648–2662, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jon Cai, Shafiuddin Rehan Ahmed, Julia Bonn, Kristin Wright-Bettner, Martha Palmer, and James H. Martin. 2023. CAMRA: Copilot for AMR annotation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 381–388, Singapore. Association for Computational Linguistics.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2021. Cross-document coreference resolution over predicted mentions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5100–5107, Online. Association for Computational Linguistics.

Prafulla Kumar Choubey and Ruihong Huang. 2017. Event coreference resolution by iteratively unfolding inter-dependencies among events. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2124–2133, Copenhagen, Denmark. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Agata Cybulska and Piek Vossen. 2013. Semantic relations between events and their time, locations and participants for event coreference resolution. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 156–163, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4545–4552, Reykjavik, Iceland. European Language Resources Association (ELRA).

Donald Davidson. 1969. *The Individuation of Events*, pages 216–234. Springer Netherlands, Dordrecht.

Pascal Denis and Jason Baldridge. 2009. Global joint models for coreference resolution and named entity classification. *Procesamiento del Lenguaje Natural*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.

Nicola Guarino, Riccardo Baratella, and Giancarlo Guizzardi. 2022. Events, their names, and their synchronic structure. *Applied ontology*, 17(2):249–283.

William Held, Dan Iter, and Dan Jurafsky. 2021. Focus on what matters: Applying discourse coherence theory to cross document coreference. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1406–1417, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kian Kenyon-Dean, Jackie Chi Kit Cheung, and Doina Precup. 2018. Resolving event coreference with supervised representation learning and clustering-oriented regularization. *arXiv preprint arXiv:1805.10985*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Lawrence Brian Lombard. 2019. *Events: A metaphysical study*. Routledge.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, page 25–32, USA. Association for Computational Linguistics.

Xiaoqiang Luo, Sameer Pradhan, Marta Recasens, and Eduard Hovy. 2014. An extension of BLANC to system mentions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 24–29, Baltimore, Maryland. Association for Computational Linguistics.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, Dublin, Ireland. Association for Computational Linguistics.

Nafise Sadat Moosavi, Leo Born, Massimo Poesio, and Michael Strube. 2019. Using automatically extracted minimum spans to disentangle coreference evaluation from boundary detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4168–4178, Florence, Italy. Association for Computational Linguistics.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. Richer event description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, pages 47–56, Austin, Texas. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Sameer Pradhan, Julia Bonn, Skatje Myers, Kathryn Conger, Tim O'gorman, James Gung, Kristin Wright-bettner, and Martha Palmer. 2022. PropBank comes of Age—Larger, smarter, and more diverse. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 278–288, Seattle, Washington. Association for Computational Linguistics.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35, Baltimore, Maryland. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

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

Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania.

Zhiyi Song, Ann Bies, Justin Mott, Xuansong Li, Stephanie Strassel, and Christopher Caruso. 2018. Cross-document, cross-language event coreference annotation using event hoppers. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ERE: Annotation of entities, relations, and events. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, Denver, Colorado. Association for Computational Linguistics.

Elizabeth Spaulding, Kathryn Conger, Anatole Gershman, Mahir Morshed, Susan Windisch Brown, James Pustejovsky, Rosario Uceda-Sosa, Sijia Ge, and Martha Palmer. 2024. PropBank goes public: Incorporation into Wikidata. In *Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII)*, pages 166–175, St. Julians, Malta. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, page 45–52, USA. Association for Computational Linguistics.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Kristin Wright-Bettner, Martha Palmer, Guergana Savova, Piet de Groen, and Timothy Miller. 2019. Cross-document coreference: An approach to capturing coreference without context. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 1–10, Hong Kong. Association for Computational Linguistics.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2023. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *arXiv preprint arXiv:2307.02477*.

Runxin Xu, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. 2022. A two-stream AMR-enhanced model for document-level event argument extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5025–5036, Seattle, United States. Association for Computational Linguistics.

Xianjun Yang, Yujie Lu, and Linda Petzold. 2022a. Few-shot document-level event argument extraction. *ArXiv*, abs/2209.02203.

Xiaohan Yang, Eduardo Peynetti, Vasco Meerman, and Chris Tanner. 2022b. What GPT knows about who is who. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 75–81, Dublin, Ireland. Association for Computational Linguistics.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. Transfer learning from semantic role labeling to event argument extraction with template-based slot querying. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2627–2647, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.