X-AMR Annotation Tool

Shafiuddin Rehan Ahmed **Jon Z. Cai** **Martha Palmer** **James H. Martin**
University of Colorado, Boulder, USA
{shah7567, jon.z.cai}@colorado.edu

Abstract

This paper presents a novel **Cross**-document **A**bstract **M**eaning **R**epresentation (X-AMR) annotation tool designed for annotating key corpus-level event semantics. Leveraging machine assistance through the Prodigy Annotation Tool, we enhance the user experience, ensuring ease and efficiency in the annotation process. Through empirical analyses, we demonstrate the effectiveness of our tool in augmenting an existing event corpus, highlighting its advantages when integrated with GPT-4. Code and annotations: github.com/ahmeshaf/gpt_coref[1] [2]

1 Introduction

Semantic representations of events play a pivotal role in natural language processing (NLP) tasks, facilitating the understanding and extraction of meaningful information from text. Among the various approaches to represent events, Semantic Role Labeling (SRL; Palmer et al. (2005)) and Abstract Meaning Representation (AMR; Banarescu et al. (2013)) have gained significant attention. In this paper, we delve into the realm of semantic event representations, with a particular focus on a method for expanding AMR.

AMR, a graph-based semantic representation, aims to capture the underlying meaning of sentences by breaking them down into atomic concepts and their semantic relationships. Each concept in AMR is associated with a unique identifier, and the relationships between concepts are represented as labeled edges in a graph. AMR has proven to be versatile, serving as a valuable resource for a wide range of NLP tasks such as machine translation, question answering (Fu et al., 2021), and summarization (Liao et al., 2018). Its ability to provide a structured, language-independent representation of textual content makes it an essential tool in the NLP toolkit.

However, despite its many merits, current AMR techniques are not without limitations. One of the primary challenges lies in linking temporal relations and entity coreference across sentences and documents. This limitation hinders the comprehensive understanding of text, as it often fails to capture the intricate interplay between events and entities that span multiple contexts. This issue becomes particularly pronounced in scenarios involving cross-document event coreference, where events mentioned in one document need to be linked to events in other documents for a coherent understanding of a larger narrative.

To illustrate the challenge of coreference across documents, consider the following example: Two news articles discuss a corporate acquisition. In one article, the event is described as "Company A's purchase of Company B on July 1st, 2008" while in another article, it is referred to as "In 7/08 Company B was acquired by Company A." Establishing the coreference relationship between these two descriptions is non-trivial, yet crucial for creating a comprehensive representation of the acquisition event.

To specifically address the intricate challenges of cross-document event coreference resolution, our research introduces two significant contributions. Firstly, we propose a novel framework X-AMR. This framework is an enhancement of the existing AMR, specifically designed to overcome the challenges inherent in linking events and entities across different documents. X-AMR effectively combines the strengths of AMR with the ability to create a more comprehensive and coherent depiction of narratives that span multiple sources.

Secondly, the development of a specialized interface is another key contribution of our work. Utilizing the model-in-the-loop annotation methodology, we have leveraged the customized Prodigy

[1]Demo: https://youtu.be/TuirftxciNE
[2]Live Link: eacldemo.acl-lawpaper34-demo.site/

annotation tool to augment an existing event coreference dataset, the Event Coref Bank plus (ECB+; Cybulska and Vossen (2014)). This development has facilitaed the annotation of X-AMR representations, focusing on the annotation interface and the enhanced X-AMR dataset. Additionaly, we present an evaluation showcasing the accuracy and efficiency of our approach. Our research endeavors to demonstrate the effectiveness of X-AMR in addressing the limitations of current sentence level AMR, especially in linking temporal relations and entity coreference across sentences and documents.

2 Related Work

AMR is a formalism meticulously crafted to capture the semantic nuances of natural language expressions with versatile and expressive power. In the field of Natural Language Processing (NLP), automatic AMR parsing transforms natural language inputs into formal AMR representations, which have demonstrated utility in a diverse array of downstream applications such as Summarization (Liao et al., 2018), Dialog systems (Bonial et al., 2020; Bai et al., 2021), Question-Answering (Kapanipathi et al., 2021), Machine Translation (Li and Flanigan, 2022), Language Modeling (Bai et al., 2022), and Fact Checking (Ribeiro et al., 2022).

Formally, AMR are structured as labeled, rooted, directed acyclic graphs, which capture abstract concepts, predicate-argument relationships, and entities found in sentences or utterances. They integrate the semantic content addressed by different representation schemes such as SRL, named entities recognition (NER; Wang et al. (2022)), and coreference resolution into a unified representation. For example, for sentence "HP acquired EYPMCS.", the corresponding AMR is:

```
(d / acquire-01
    :ARG0 (c / company
        :name (n / name
            :op1 "HP"))
    :ARG1 (c2 / company
        :name (n2 / name
            :op1 "EYPMCS"))
```

The above AMR graph captures concepts such as events such as "acquire", named entities such as the HP company, and properties of the entity such as their names as graph nodes and subgraphs. Their interrelations between concepts and events are then depicted through labeled edges. Events are denoted using Propbank rolesets, and semantics relations of the entities and events are specified through numbered arguments and non-core relations from AMR's role inventory. For example, in the above acquisition event, the `ARG0` typically specifies the stereotypical agent of an event and `ARG1` typically specifies the stereotypical patient of an event. Additionaly, AMR graphs formalize local temporal information, as shown in the provided example.

In the preceding disucssion, we highlighted the expressiveness of AMR. However, the expressiveness of AMR introduces complexities in AMR annotation, historically a significant bottleneck for NLP community. The challenge has been to provide a substantial volume of AMR annotations to the data hunger statistical machine learning models given the limitations of available tools. The ISI editor, serving as the first AMR editor, has supported the AMR community for over a decade. Despite the efficacy of the ISI editor, its learning curve is notably steep for annotators. To make AMR annotation more accessible, Cai et al. (2023) developed a new annotation approach. They introduced an AMR editor based on coding, complemented by a neural network parser model, to streamline the annotation process.

The remarkable progress in large language model-based coding assistance, pioneered by OpenAI and Microsoft, is transforming the landscape of program synthesis in software engineering. These models, trained in both natural language and programming languages, excel at completing programs by intelligently integrating code history and human instructions. In a similar vein, CAMRA leverages these large language models (LLMs) to enhance AMR annotation. We are pioneering the extension of LLMs' capabilities, broadening their application to include more complex tasks such as cross-sentential and cross-document coreference and event linking. This initiative represents a significant step forward in harnessing the power of LLMs for even more sophisticated and long-distance dependent language processing tasks.

3 Annotation Methodology

The annotation workflow, as depicted in Figure 1, comprises of two phases. In the first phase we annotate the roleset IDs of the event triggers. Then we specify the arguments of the event incrementally. During these two phases, we maintain an arguments store and a model-in-the-loop that queries the store and suggests annotators with the most

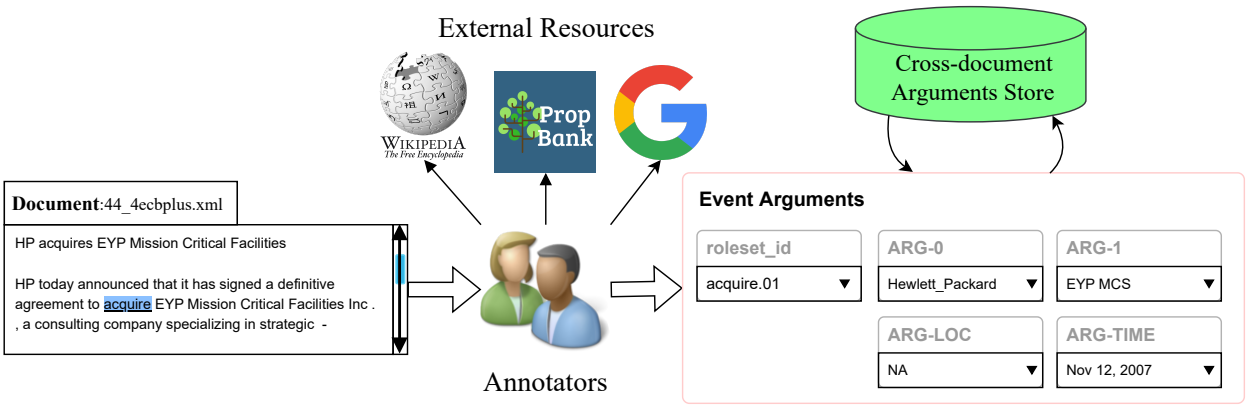

Figure 1: The Annotation Methodology of X-AMR. The annotators are presented with PropBank and are allowed to use external resources, such as Wikipedia and Google News, during the annotations.

likely arguments. This store and the model are updated when new events are annotated.

Next, we discuss the annotation guidelines, the interface, and the model-in-the-loop in the annotation workflow.

3.1 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 a model-in-the-loop annotation methodology with the prodi.gy annotation tool.

3.1.1 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*).

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 2 gives an example Frame File for

```
agree.01 - agree    agree.01
 ARG-0: Agreer       ARG-0: HP
 ARG-1: Proposition  ARG-1: acquire.01
                        ARG-1: EYP
```

Figure 2: The PropBank roleset definitions of agree.01 and the expected annotations in X-AMR.

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[3] or Maven (Wang et al., 2020).

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

Figure 3: Eventive ARG-1 in the roleset agree.01. The ARG-1 clause is annotated as the connecting event with roleset ID acquire.01

3.1.2 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.

3.1.3 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.

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 3 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.

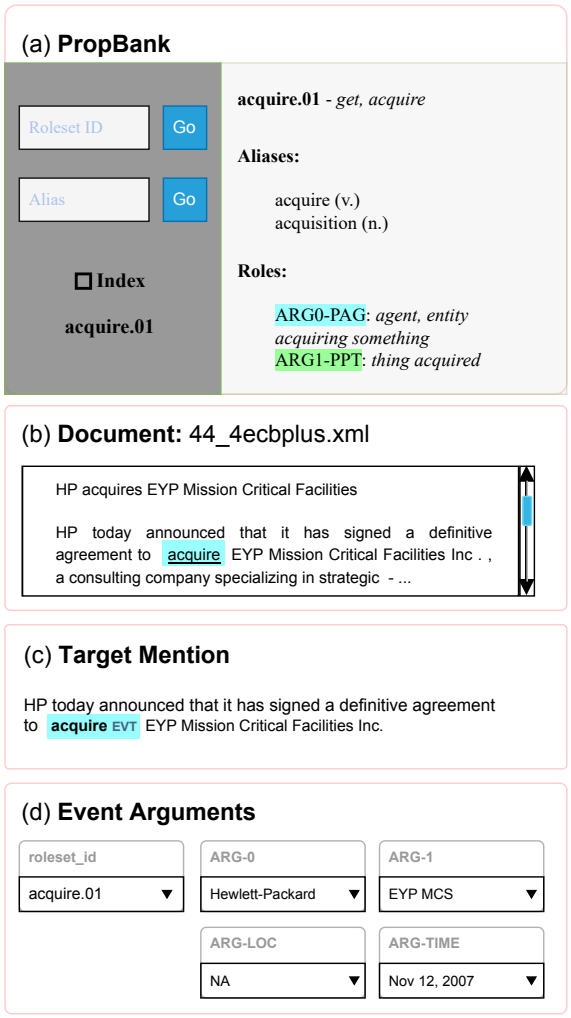

Figure 4: The Annotation Interface Using prodi.gy Annotation Tool

ARG-Loc & ARG-Time Identification Annotators may also utilize external resources, such as Wikipedia[4], 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.

3.2 Annotation Interface

The annotation interface, as depicted in Figure 4, comprises four distinct components: (a) the integrated PropBank website, (b) the document view, (c) the sentence view, and (d) the event argument forms. This interface is hosted on a server using Prodigy, with links distributed to individual annotators.

PropBank Website: We adapt the publicly avail-

[4]Although we add this in the guidelines, the annotators do not wikify. Our choice is to use Wikipedia over the more commonly used KB-wikidata because of GPT-friendly identifiers of the pages. Check out Appendix B.

able PropBank website builder[5] to ensure compatibility within an embedded environment. This interactive website hosts an indexed list of roleset definitions that annotators refer to.

Target Mention Document: The document containing the current mention is fully displayed in a scrollable view with the event trigger highlighted upon interface loading, facilitating easy access to additional context for annotators.

Target Mention Sentence: This section displays the sentence encompassing the mention, with the event trigger highlighted in Prodigy's named entity recognition (NER) style. Typically, a sentence alone is sufficient to identify the arguments, and therefore, it is in the field of focus first.

Event Arguments Forms: The event argument forms are located in this section, enabling annotators to manually input corpus-level arguments for the events. Each form is equipped with a dropdown list containing previously annotated arguments, facilitating the annotation process. Figure 5 shows the different kinds of arguments stored in each of the argument forms. The `roleset_id` form stores all the rolesets in PropBank, `ARG-0` and `ARG-1` the identified agents and patients up til then, `ARG-LOC` the locations, and `ARG-Time` the dates.

3.3 Model-in-the-loop

Incorporating a model-in-the-loop approach, our annotation framework utilizes a straightforward Word2Vec classifier implemented using spaCy. This classifier ranks sentences containing previously seen arguments in relation to the target sentence. The dynamic ranking of these sentences is reflected in the dropdown list, with the highest-ranked sentence positioned at the top. The annotator is presented with the option to either accept or reject the top-ranked arguments.

Argument Ranking and Selection: Upon loading the annotation interface, the system ranks the arguments from previously annotated sentences alongside the target sentence. The highest-ranked argument is selected by default and presented as the initial choice to the annotator. This ranking is based on the similarity or relevance of the sentences as determined by the Word2Vec classifier.

Acceptance and Integration: Should the annotator choose to accept the top-ranked sentence, it is seamlessly integrated into the set of previous arguments. This integration enhances the corpus-level annotation by incorporating contextually relevant information from the selected sentence.

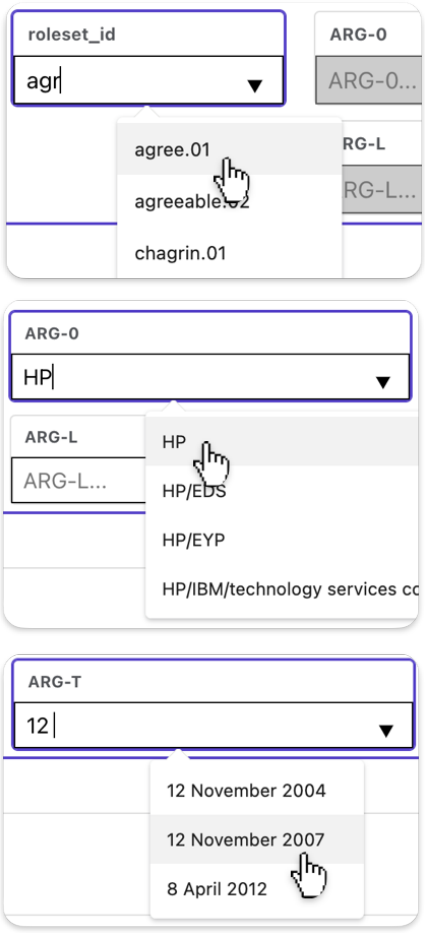

Figure 5: Screenshots

Rejection and New Argument Creation: In the event of rejection, the system generates new arguments, leveraging the embedding of the rejected sentence. This adaptive mechanism ensures that even when an annotator rejects the top-ranked sentence, valuable information is not lost. Instead, it is used to generate potentially relevant arguments for further annotation.

GPT-in-the-loop: Finally, yet importantly, we employ a GPT-based methodology to streamline the extraction of cross-document arguments through a two-step Retrieval Augmented Generation process. A comprehensive breakdown of our prompt engineering techniques is provided in Appendix B. The primary objective of this approach is to establish cross-document entity coreference.

Because of budget constraints, we have limited the execution of this experiment to a subset of the Dev dataset (Dev-small), encompassing a total of 120 mentions. Corpus statistics and annotation

[5]https://github.com/propbank/propbank-frames

analysis are detailed in Appendix A.

4 Analysis

4.1 Model-in-the-loop

We collect X-AMR annotations on the ECB+ dataset, as detailed in Appendix A (refer to the appendix for specific numerical data and human annotation analysis). During the annotation process, we collect human annotations along with predicted rolesets and arguments generated by our model. We assess the model's performance by comparing its predictions to human annotations. We carefully recorded the instances in which annotators made modifications to the predicted text provided by the model. We count the acceptance ratio of the predictions, which not only signifies the model's effectiveness but also represents the amount of effort saved by annotators.

Our analysis on the train, dev, and test sets of ECB+, as illustrated in Figure 7, reveals several noteworthy observations: the correct roleset ID prediction consistently exceeded 80% for both annotators, denoted as A1 and A2. A1 appeared to be more inclined to accept the model's argument predictions compared to A2. This experiment serves as a foundation for future research, and one potential avenue is to incorporate these findings into downstream tasks, such as Event Coreference Resolution, to evaluate the quality of annotations and explore further implications of using model-in-the-loop for X-AMR annotations.

4.2 GPT-in-the-loop

In our GPT experiment on Dev-small, we had an adjudicator review 120 mentions and note when they had to adjust GPT's predictions. The outcomes of this evaluation are visually represented in Figure 6, which illustrates the ratio of mentions requiring modification. The main takeaway here is that GPT performed well in generating Location and Time arguments but struggled with predicting roleset IDs and ARG-0, ARG-1 arguments. We believe that integrating the model-in-the-loop approach could help improve performance compared to just using GPT.

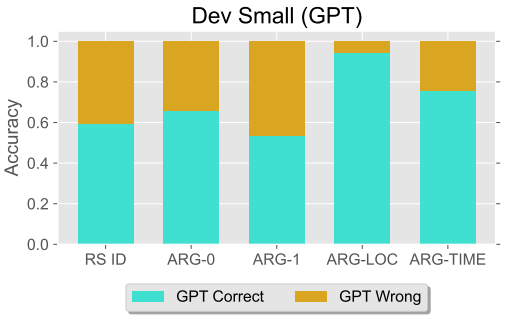

Figure 6: Accuracy of GPT Predictions of Roleset and ARG based on the gold standard annotation (adjudicated);

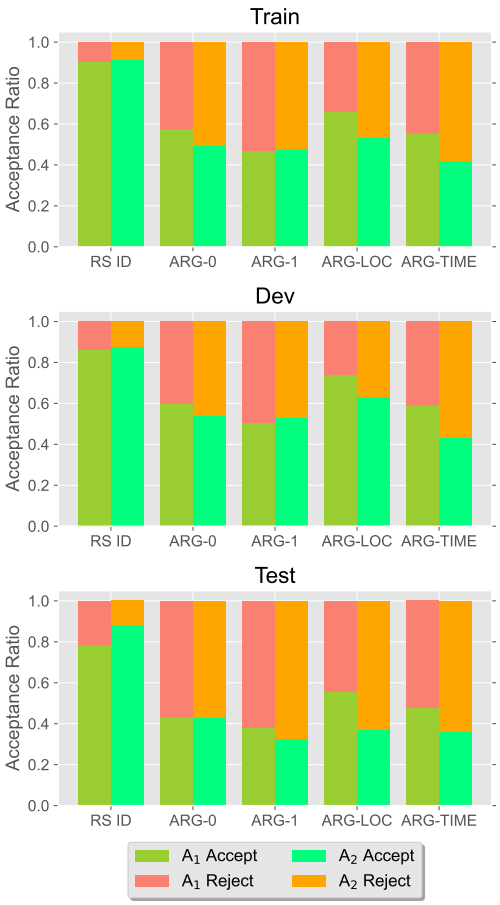

Figure 7: Roleset and ARG Analysis for A_1 and A_2: "A_x Accept" represents the acceptance rate of the model suggestions according to Annotator x; "A_x Reject" represents the rejection rate of the model suggestions according to Annotator x;

5 Future Work

The next steps include leveraging the X-AMR structures in creating efficient methods for neuro-symbolic event coreference resolution (ECR). For example, the X-AMR annotations could help in filtering the most pertinent event pairs that can be used with more resource intensive methods for estimating coreference (Ahmed et al., 2023a). Another important direction is in the estimation of the quality and cost savings of our methodology in doing

ECR annotations. Quality measured by the number of ECR links that can be found with the least amount of pairwise event mention comparisons (Ahmed et al., 2023b).

6 Conclusion

In this paper, we have introduced a novel approach for cross-document, corpus-level semantic event extraction utilizing the X-AMR framework. To facilitate this process, we have developed a model-in-the-loop annotation tool tailored for X-AMR annotation, seamlessly integrated with Prodigy. This tool has been employed to curate X-AMR annotations by enriching an existing event coreference dataset, with contributions from two annotators. To evaluate the effectiveness of our approach, we have introduced a comprehensive assessment of the predictions, incorporating both the model's output and the assistance of GPT.

Limitations

This work has several limitations. Firstly, the annotation tool used is a one-time paid software, which may restrict its accessibility to some researchers, although we have made the annotation recipe freely available. Secondly, the study relies on gold mentions rather than predicted ones, suggesting a need for future research to incorporate an additional annotation process to identify event triggers. Lastly, the non-reproducibility of GPT is acknowledged, and it may have been pre-trained on the corpus. However, we provide GPT-generated outputs and use them primarily for information generation rather than prediction, especially in event description generation. Future work may focus on distilling information into smaller, reproducible models to address these limitations and enhance the robustness of our approach.

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 EACL 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

The authors would like to thank the reviewers of EACL 2024 System Demonstrations 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, 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.

References

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.

Xuefeng Bai, Yulong Chen, Linfeng Song, and Yue Zhang. 2021. Semantic representation for dialogue

modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4430–4445, Online. Association for Computational Linguistics.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

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.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

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.

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).

Ruiliu Fu, Han Wang, Xuejun Zhang, Jun Zhou, and Yonghong Yan. 2021. Decomposing complex questions makes multi-hop QA easier and more interpretable. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 169–180, Punta Cana, Dominican Republic. 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.

Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernandez Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, Dinesh Garg, Alfio Gliozzo, Sairam Gurajada, Hima Karanam, Naweed Khan, Dinesh Khandelwal, Young-Suk Lee, Yunyao Li, Francois Luus, Ndivhuwo Makondo, Nandana Mihindukulasooriya, Tahira Naseem, Sumit Neelam, Lucian Popa, Revanth Gangi Reddy, Ryan Riegel, Gaetano Rossiello, Udit Sharma, G P Shrivatsa Bhargav, and Mo Yu. 2021. Leveraging Abstract Meaning Representation for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3884–3894, Online. Association for Computational Linguistics.

Changmao Li and Jeffrey Flanigan. 2022. Improving neural machine translation with the Abstract Meaning Representation by combining graph and sequence transformers. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*, pages 12–21, Seattle, Washington. Association for Computational Linguistics.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract Meaning Representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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

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.

Leonardo F. R. Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, and Mohit Bansal. 2022. FactGraph: Evaluating factuality in summarization with semantic graph representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3238–3253, Seattle, United States. 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.

Yu Wang, Hanghang Tong, Ziye Zhu, and Yun Li. 2022. Nested named entity recognition: A survey. *ACM Trans. Knowl. Discov. Data*, 16(6).

A Dataset Details

The ECB+ corpus is a popular English corpus used to train and evaluate systems for event coreference resolution. It extends the Event Coref Bank corpus (ECB; Bejan and Harabagiu (2010)), with annotations from around 500 additional documents. The corpus includes annotations of text spans that represent events, as well as information about how those events are related through coreference. We divide the documents from topics 1 to 35 into the training and validation sets2 , and those from 36 to 45 into the test set, following the approach of Cybulska and Vossen (2014).

A.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.

	Train	Dev	Test	Dev small
Documents	594	196	206	91
Mentions	3808	1245	1780	120
Roleset ID Agreement	0.84	**0.91**	0.80	–
w/ X-AMR	3195*	1245	847*	120
w/ Nested ARG-1	1081	325	220	24
w/ ARG-Loc	2949	1243	707	120
w/ ARG-Time	3192	1244	805	120

Table 1: ECB+ 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.

B Prompt Engineering

Our approach for X-AMR extraction with GPT involves a two-step process. In the initial step, we extract the Event Description along with the document-level arguments of the event by utilizing prompts such as `Instructions A`, `JSON Labels A`, and `Inputs A`. Following the generation of individual event descriptions through this step, we employ another prompt-based technique to generate corpus-level arguments.

In this secondary method, we introduce an additional instruction into `Instructions A`, forming `Instructions B`. This instruction directs GPT to identify the most informative Event Description that is coreferent with the current Event. Subsequently, we provide this identified Event Description (`JSON Labels B`) within the context and task GPT with generating missing information, such as date and location, pertaining to the target event. We provide the list of informative event descriptions in the topic of the target event in `Inputs B`.

The estimated cost of running this experiment is about $15.

Instructions A

```
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.
```

JSON Labels A

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

Inputs A

Target Mention Document: Entire document with the marked event trigger

Target Mention Sentence: Sentence with the marked event trigger

Instructions B

Instructions A

6. Identify the most informative (having Wikipedia and complete dates) and best matching Event Description from the provided list of descriptions.

JSON Labels B

JSON Labels A

Most Informative Event Description: Pick the most informative event description from the Event Description List. Choose by selecting the one that has complete date and Wikipedia links for the arguments and also is coreferent with the target Event. Hint: choose the one starts starts with "On DATE"

Inputs B

Event Description List: Event descriptions of the three most informative and similar events in the corpus.

Target Event Description: Event description of the target event

Target Mention Sentence: Sentence with the marked event trigger