LiDARR: Linking Document AMRs with Referents Resolvers

Jon Z. Cai* **Kristin Wright-Bettner*** **Zekun Zhao**†
Shafiuddin Rehan Ahmed* **Abijith Trichur Ramachandran***
Jeffrey Flanigan† **Martha Palmer*** **James H. Martin***
*University of Colorado Boulder †University of California Santa Cruz
jon.z.cai@colorado.edu

Abstract

In this paper, we present LiDARR (**Li**nking **D**ocument **A**MRs with **R**eferents **R**esolvers)[1], a web tool for semantic annotation at the document level using the formalism of Abstract Meaning Representation (AMR). LiDARR streamlines the creation of comprehensive knowledge graphs from natural language documents through semantic annotation. The tool features a visualization and interactive user interface, transforming document-level AMR annotation into an models-facilitated verification process. This is achieved through the integration of an AMR-to-surface alignment model and a coreference resolution model. Additionally, we incorporate PropBank rolesets into LiDARR to extend implicit roles in annotated AMR, allowing implicit roles to be linked through the coreference chains via AMRs.

1 Introduction

Abstract Meaning Representation (AMR) has become one of the most extensively used semantic representation formalisms in the field of Natural Language Processing (NLP). It effectively captures the lexical semantics of natural language text by resolving predicative relationships, grounded in Neo-Davidsonian semantics (Banarescu et al., 2013). This process, known as AMR parsing, allows us to answer fundamental questions such as "who did what to whom, when, where, and how," while also addressing complex ontological relationships between various concepts. AMR's transparent symbolic representation of natural language makes it particularly valuable for AI applications that require semantic inference and interpretability.

An example of Multi-sentence AMR (MS-AMR) is illustrated in Figure 1, which shows an AMR graph for the sentences: "The boy wants the girl to believe him. Yet, she doesn't believe him." In the graph of the first sentence, "want" acts as the primary predicate, and the desire agent to be "boy" and the desired entity to be the "believe" state predicate. Such a structure can be queried using graph query languages like SPARQL (Prud'hommeaux and Seaborne, 2008) and Cypher (Francis et al., 2018) with minimal adaptation.

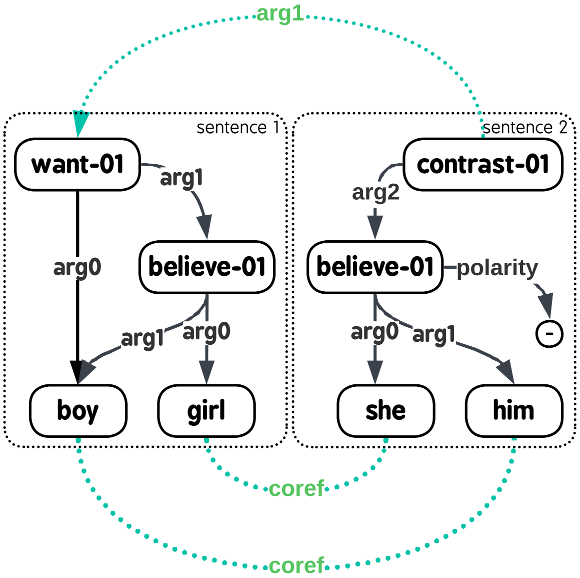


Figure 1: AMR for sentences "*the boy wants the girl to believe him. Yet, she doesn't believe him*" in conventional graph representation format; green dotted edges denote cross sentence coreference links and implicit argument links, which are MS-AMR specific

The more compact but equivalent PENMAN encoding (Goodman, 2019, 2020) of the two single sentence AMRs are:

```
(w / want-01           |(c / contrast-01
 :ARG0 (b / boy)       | :ARG1 (b2 / believe-01
 :ARG1 (b1 / believe-01|  :ARG0 (s / she)
  :ARG0 (g / girl)     |  :ARG1 (h / him)))
  :ARG1 b))            |
```

In the context of data-driven machine learning, researchers have annotated tens of thousands of natural-language-AMR pairs. These annotations

---

[1] demo video: `https://youtu.be/Ab32NEEA90U`; tool available at: https://camera.colorado.edu/docview2

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 426–435
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

enable the training of advanced deep learning-based parsers and facilitate extensive quantitative evaluations of semantic understanding. While AMR is capable of resolving semantics regardless of text length in theory, practical annotations are typically limited to single sentences or small sentence clusters due to the increasing complexity of larger AMR graphs. This limitation results in rich semantic graphs being isolated rather than forming a unified network at the document level. To maximize the potential of AMR, it is essential to integrate these sentence-level graphs into a coherent semantic network through coreference resolution.

Coreference resolution involves identifying and grouping different expressions that refer to the same entity. For instance, in the example sentences in Figure 1, both “girl” and “she” refer to the same entity and are considered coreferences. Effective coreference resolution is crucial for intelligent systems, as it requires a profound understanding of semantics and world knowledge. It is particularly important for tasks such as navigating large text corpora and ensuring the consistency and reliability of high-stakes documents like legal and medical records. Ultimately, integrating coreference resolution with AMR allows the creation of a cohesive document-level representation from isolated sentence-level semantic graphs.

The challenges of integrating coreference information into sentence-level AMRs lie in two main areas. First, AMR graphs are often coded without explicit alignment between the surface text and the corresponding nodes and edges, making the alignment mapping complex to produce. Second, document-level AMRs require implicit roles to be part of the coreference chain, which is not feasible using only the surface text, necessitating an annotation interface that works directly on the AMRs. Current annotation tools, such as Anafora and UMR Writer, rely heavily on direct annotation of the AMR structures. However, AMRs are less intuitive to comprehend than surface text, and the lack of facilitation for coreference in AMR makes the task even more challenging.

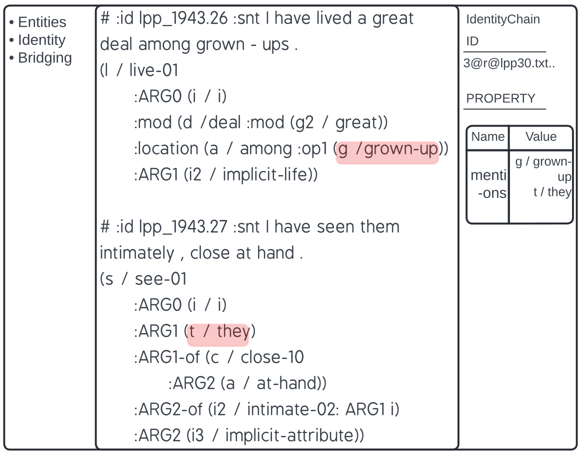
Our design addresses these challenges, and we summarize our contributions as follows:

- **Integration of Alignment Models:** We incorporated state-of-the-art alignment models to provide initial suggestions for aligning surface text with AMR nodes. This results in a quality control process during alignment annotation.
- **Coreference Resolution Models:** We integrated coreference resolution models to provide initial suggestions for coreference clustering. By calculating the overlap of mentions with the alignment spans from the first step, we formed coreference clusters among AMRs within a document.
- **Customized Interface:** We designed a novel, customized, and dynamic interface to facilitate simultaneous navigation of the text and AMRs, making document-level AMR annotation a clustering correction task.

The modular design of our system ensures that it is easily extensible and adaptable to more advanced models, such as Large Language Models (LLMs), enhancing its capability and usability.

2 Related Work

Anafora (Chen and Styler, 2013) and UMR Writer (Zhao et al., 2021) are the two primary tools currently supporting document-level AMR annotation. Anafora’s extension for document-level AMR annotation was introduced by O’Gorman et al. by replacing the regular text in the Anafora interface with AMRs represented in PENMAN encoding. In this setup, annotating coreference among AMRs involves specifying mention spans directly in the AMR code. Figure 2 illustrates this interface.



```
# :id lpp_1943.26 :snt I have lived a great
deal among grown - ups .
(l  / live-01
      :ARG0 (i  / i)
      :mod (d  /deal :mod (g2  / great))
      :location (a  / among :op1 (g  /grown-up))
      :ARG1 (i2  / implicit-life))

# :id lpp_1943.27 :snt I have seen them
intimately , close at hand .
(s  / see-01
      :ARG0 (i  / i)
      :ARG1 (t  / they)
      :ARG1-of (c  / close-10
            :ARG2 (a  / at-hand))
      :ARG2-of (i2  / intimate-02: ARG1 i)
      :ARG2 (i3  / implicit-attribute))
```

Name	Value
menti-ons	g / grown-up t / they

Figure 2: Demonstration of the Document AMR annotation interface within Anafora

While this approach allows for the annotation of document-level AMRs using other coreference tools, it also highlights a key challenge: the need for flexible span selection. Annotating arbitrary text spans in natural language text requires that users can select any span in the interface. However, because AMR graphs are encoded in PENMAN encoding with a well-defined syntax, this flexibility can become a hindrance rather than a help. Annotators must carefully manage span selection, which can be cumbersome.

UMR Writer is another tool capable of annotating document-level AMRs. The annotation process in UMR Writer mirrors that of regular AMR annotation. Usefully users can create standalone document level graphs that group all coreferent concepts as `:coref` roles in the document level graph. However, due to the pairwise selection process this method is tedious, not an ideal solution for creating comprehensive document-level AMRs.

Moreover, both Anafora and UMR Writer localize sentence-AMR pairs, limiting the flexibility of navigating each representation independently. This constraint can impose a cognitive burden on annotators compared to reading natural language text alone. Even for highly experienced AMR experts, natural language text remains more familiar and frequently encountered than AMRs, making the latter a less preferred medium for annotation tasks.

A more recent tool, CAMRA (Cai et al., 2023), designed for annotating sentence-level AMRs, also holds potential for coreference annotation similarly to UMR Writer. CAMRA features a quick, click-based alignment interface that allows annotators to specify the alignment between surface text and AMR nodes, making it possible to work more on the surface text like other coreference annotation tools. However, CAMRA's single sentence UI makes it challenging to fit long MS-AMR content and navigate among mention clusters.

A closely related tool, X-AMR (Ahmed et al., 2024), focuses on cross-document event coreference annotation, addressing the specific challenge of linking events across documents. INCEpTION (Klie et al., 2018) and WebAnno (Eckart de Castilho et al., 2016) offer broader functionality, including entity linking at the surface, which may support Semantic Role Labeling (SRL) enrichment but is less suited for configuring deeper semantic representations such as AMR.

Inspired by the strengths and limitations of these tools, our work aims to combine their features organically to provide a more modern and streamlined user experience for document-level AMR annotation. Our approach integrates state-of-the-art alignment models to suggest alignments between surface text and AMR nodes, coreference resolution models to form coreference clusters, and a dedicated interface to navigate text and AMRs flexibly. This results in a cohesive system that simplifies document-level AMR annotation, making it more efficient and user-friendly.

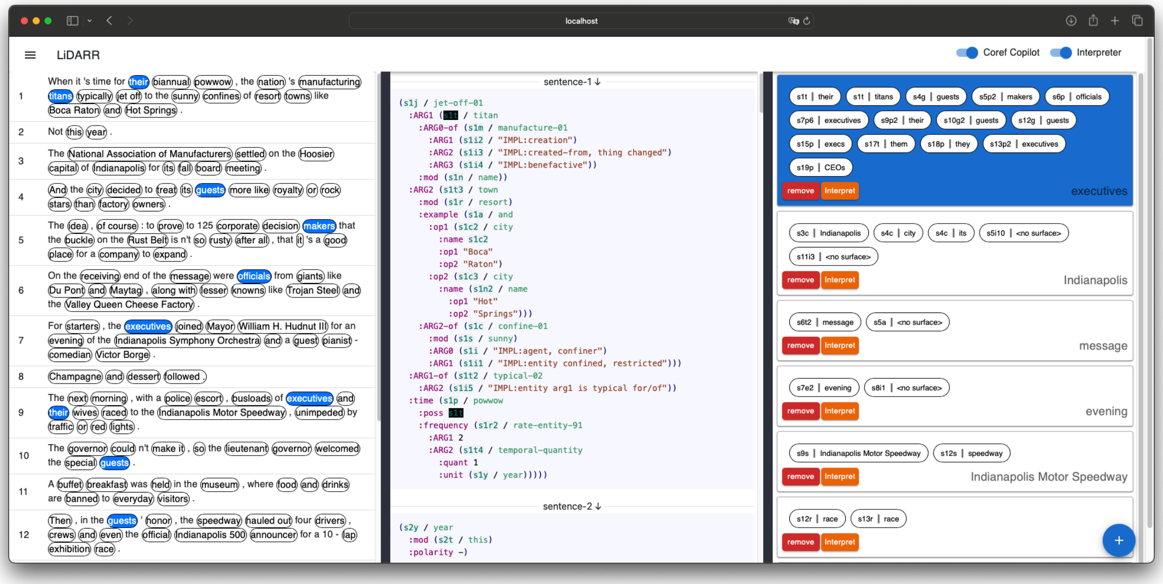
3 System Design and Features

Constructing document-level AMRs presents unique challenges due to the necessity of linking long-distance references within the text and the significant cognitive load on annotators. This task is akin to sorting a deck of cards by suit; the more shuffled the deck, the more challenging the sorting process becomes. The complexity of annotating coreferences makes it particularly helpful to integrate existing models to create even partially sorted clusters, thereby easing the annotators' workload. Incorporating AMR adds another layer of complexity, requiring a cohesive alignment that merges coreference cluster information with AMR nodes. We designed the Annotation User Interface (AUI) with the following core requirements:

- **Rendering Surface Text and AMR**: The AUI must display both the surface text and AMR, with coreference annotations performed primarily on the surface text to leverage trained coreference resolution models.
- **Linking Mentions to AMR Nodes**: Mentions in the surface text should be linked to AMR nodes by calculating overlaps between spans produced by the AMR-surface alignment model and the coreference resolution model. This ensures that grouping surface mentions induces the grouping of AMR concept nodes.
- **Handling Implicit Mentions**: Annotators should be able to include AMR nodes that do not have a surface correspondence to account for implicit mentions.
- **Intuitive Visualization**: The AUI should clearly indicate clusters in the text and AMRs through visualizations.
- **Model Assistance**: The invocation of AI assistance should be automatic yet controllable by the user, ensuring convenience and privacy awareness.

3.1 Features in User Interface

We show an overview of the Annotation User Interface of LiDARR in Figure 3. Inspired by the



1 When it 's time for their biannual powwow , the nation 's manufacturing titans typically jet off to the sunny confines of resort towns like Boca Raton and Hot Springs .
2 Not this year .
3 The National Association of Manufacturers settled on the Hoosier capital of Indianapolis for its fall board meeting .
4 And the city decided to treat its guests more like royalty or rock stars than factory owners .
5 The idea , of course : to prove to 125 corporate decision makers that the buckle on the Rust Belt is n't so rusty after all , that it 's a good place for a company to expand .
6 On the receiving end of the message were officials from giants like Du Pont and Maytag , along with lesser knowns like Trojan Steel and the Valley Queen Cheese Factory .
7 For starters , the executives joined Mayor William H. Hudnut III for an evening of the Indianapolis Symphony Orchestra and a guest pianist - comedian Victor Borge .
8 Champagne and dessert followed .
9 The next morning , with a police escort , busloads of executives and their wives raced to the Indianapolis Motor Speedway , unimpeded by traffic or red lights .
10 The governor could n't make it , so the lieutenant governor welcomed the special guests .
11 A buffet breakfast was held in the museum , where food and drinks are banned to everyday visitors .
12 Then , in the guests ' honor , the speedway hauled out four drivers , crews and even the official Indianapolis 500 announcer for a 10 - lap exhibition race .

sentence-1 ↓

```
(s1j / jet-off-01
  :ARG1 (s1t / titan
    :ARG0-of (s1m / manufacture-01
      :ARG1 (s1i2 / "IMPL:creation")
      :ARG2 (s1i3 / "IMPL:created-from, thing changed")
      :ARG3 (s1i4 / "IMPL:benefactive"))
    :mod (s1n / name))
  :ARG2 (s1t3 / town
    :mod (s1r / resort)
    :example (s1a / and
      :op1 (s1c2 / city
        :name s1c2
        :op1 "Boca"
        :op2 "Raton")
      :op2 (s1c3 / city
        :name (s1n2 / name
          :op1 "Hot"
          :op2 "Springs")))
    :ARG2-of (s1c / confine-01
      :mod (s1s / sunny)
      :ARG0 (s1i / "IMPL:agent, confiner")
      :ARG1 (s1i1 / "IMPL:entity confined, restricted")))
  :ARG1-of (s1t2 / typical-02
    :ARG2 (s1i5 / "IMPL:entity arg1 is typical for/of"))
  :time (s1p / powwow
    :poss s1t
    :frequency (s1r2 / rate-entity-91
      :ARG1 2
      :ARG2 (s1t4 / temporal-quantity
        :quant 1
        :unit (s1y / year)))))
```

sentence-2 ↓

```
(s2y / year
  :mod (s2t / this)
  :polarity -)
```

Figure 3: an overview of the main Annotation User Interface of LiDARR. Cluster for the "executives" mention has been clicked and activated. Text spans highlighted with blue background color indicates they are coreferences of the referent “executives” entity. The corresponding AMR nodes are marked with dark gray background color.

design of CAMRA, with three horizontally parallel panels: the Text Panel, AMR Panel, and Cluster Panel.

Text Panel: The text panel renders the entire document in a two-column table, with the first column showing the sentence index for easy reference and the second column displaying the sentence. Each sentence has pre-specified mention spans that are clickable for inclusion in a cluster. The sentence index cell serves as a quick navigation point to bring the corresponding AMRs to the center of the AMR panel. This design emphasizes the compactness of text rendering, mimicking the familiar typeface of natural text while providing easy access to AMR navigation.

AMR Panel: The middle panel renders individual sentence-level AMRs using PENMAN encoding, an encoding language widely adopted among annotators. Each variable in the graph is clickable, similar to the mention spans in the text panel. Clicking on an AMR variable allows annotators to include or exclude nodes in a cluster. The AMRs undergo preprocessing to fill implicit roles for each predicate according to the PropBank roleset (Palmer et al., 2005; Pradhan et al., 2022), providing anchors for implicit concepts. This capability to link implicit roles distinguishes document-level AMR parsing from standard coreference tasks. For example, in the sentences *“Taylor ended up flying with Alaska Airlines. She was compensated with a coupon after she arrived in New York,”* the predicate “fly-01” in the first sentence has an agent role (the pilot), a patient role(the passenger, Taylor), and destination role(New York). Although the destination is implicit in the first sentence, it becomes explicit in the second, allowing for linking through AMR, which is difficult on the surface form.

Clusters Panel: The rightmost panel presents cluster information, organizing coreferent mentions into card components labeled with the first selected surface span serving as the referent. Clicking on a cluster card activates editing mode, highlighting corresponding mentions in both the surface text and AMR panels. Annotators can add or remove spans from clusters by selecting unassigned spans or deselecting already included ones. Additionally, the label of each cluster card is editable through a right-click on the name text, which opens a pop-up text field for entering a user-defined name. Finally, we dedicated a separate but similar view for bridging clusters constructions.

Interactive Mode: We designed two UI modes to accommodate different user preferences for the copilot’s behavior: static and interactive. In static mode, the system processes the document and AMRs, then generates a clustering for users to correct. In interactive mode, it produces the same clustering but highlights the next possible token in the text panel, allowing users to lead cluster construction. This local suggestion always matches the most overlapping cluster and provides relevant recommendations accordingly.

3.2 Copilot Support

The goal of LiDARR is to make coreference data collection intuitive and efficient, which requires substantial AI support. There is limited support for Doc-AMR parsers due to limited document level AMR annotation and training. We instead merge the power of coreference resolution models on the surface text and transfer the clustering to the corresponding AMR concepts through AMR-surface alignment prediction. We show the pipeline in Figure 4. The diagram illustrates the collaborative workflow of LiDARR's copilot models for document-level AMR annotation. On the left, a document with sentences annotated in AMR is displayed. LiDARR first aligns surface tokens with corresponding AMR nodes using the alignment copilot. In the diagram, highlighted tokens and AMR nodes on the same row indicate successful alignment (only AMR nodes within the same cluster are highlighted for demonstration).

Next, a coreference resolution model is applied to the surface text, forming mention clusters (only one cluster is shown for clarity). Finally, LiDARR calculates span overlaps and transfers the mention clusters to the corresponding AMR concepts. As a result, previously distinct AMR concept nodes are unified, appearing in the same color to reflect their identity relation.

Specifically, the backend of LiDARR is equipped with a state-of-the-art AMR-surface-text alignment model, LEAMR aligner (Blodgett and Schneider, 2021). This model minimizes the effort needed to create alignments from scratch and need only verify and correct alignments, assuming the alignment map is nearly perfect, which can be done with the CAMRA tool.

Additionally, LiDARR includes a fast coreference model that processes the document text and outputs mention clusters. By performing an overlap check between spans produced by the coreference and alignment models, we attach AMR concept nodes to mentions in clusters. Given the density of alignment spans compared to mention spans, it is rare to find surface spans without attached AMR concepts for non-functional tokens.

Initial user feedback highlights the value of gradually building clusters and resolving bridging relations to help annotators internalize complex entity relationships. Copilot-generated coreference links, lacking clear explanations, can be confusing—especially when AMR concepts are mis-clustered. To address this, we integrate an LLM-based interpreter copilot and provide a configurable interface for users to set their preferred LLM API endpoint, enhancing human-AI collaboration. Details of this feature are available in Appendix A.

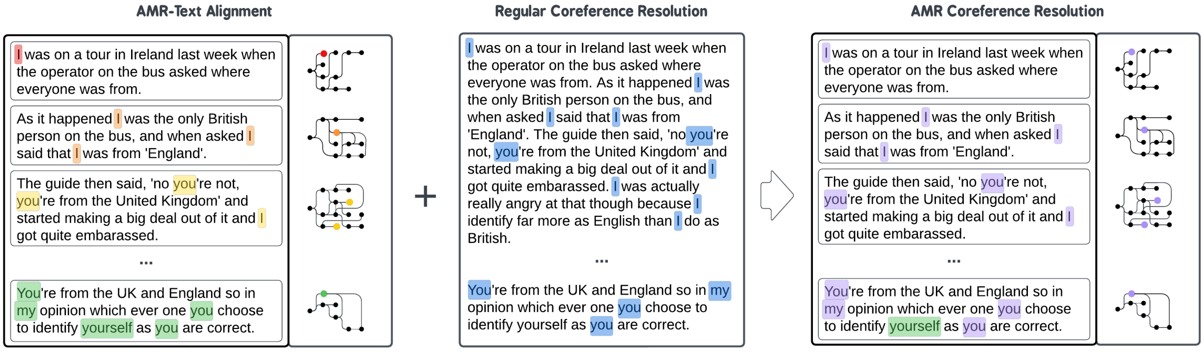
The backend uses a modular architecture, with the alignment and coreference copilots deployed as standalone REST API servers. An intermediary manager server handles data flow and communication, forming a star-shaped topology that delegates intensive tasks to dedicated servers and supports model replacement as needed.

4 Evaluation

Given that LiDARR provides AI assistance through preprocessing, the primary factor influencing user experience is accuracy. The accuracy of coreference resolution is primarily affected by the nature of the document; complex documents with frequent long-distance coreferences are naturally more challenging to resolve accurately.

We present a case study evaluating the performance of our document-level AMR annotation facilitation system, focusing on coreference resolution at the AMR concept level. While coreference resolution on surface text serves as an intermediary process, our primary objective is to facilitate coreference resolution for document-level AMR annotation. To this end, we assess system performance against a gold-standard AMR concept reference using the test set of the MS-AMR corpus (O'Gorman et al., 2018) for the ease of AMR concept coreference on this dataset. The MS-AMR test split contains nine documents annotated with MS-AMR graphs. This corpus provides annotation for identical clusters, set-membership and part-whole relations between AMR concepts. The evaluation is conducted on the identical clusters.

Mention based metric: A well-known formulation for the minimal number of mention reassignments required to convert the system's clustering S into the gold clustering G (over the same mention



I was on a tour in Ireland last week when the operator on the bus asked where everyone was from.

As it happened I was the only British person on the bus, and when asked I said that I was from 'England'.

The guide then said, 'no you're not, you're from the United Kingdom' and started making a big deal out of it and I got quite embarassed.

...

You're from the UK and England so in my opinion which ever one you choose to identify yourself as you are correct.

I was on a tour in Ireland last week when the operator on the bus asked where everyone was from. As it happened I was the only British person on the bus, and when asked I said that I was from 'England'. The guide then said, 'no you're not, you're from the United Kingdom' and started making a big deal out of it and I got quite embarassed. I was actually really angry at that though because I identify far more as English than I do as British.

...

You're from the UK and England so in my opinion which ever one you choose to identify yourself as you are correct.

I was on a tour in Ireland last week when the operator on the bus asked where everyone was from.

As it happened I was the only British person on the bus, and when asked I said that I was from 'England'.

The guide then said, 'no you're not, you're from the United Kingdom' and started making a big deal out of it and I got quite embarassed.

...

You're from the UK and England so in my opinion which ever one you choose to identify yourself as you are correct.

Figure 4: LiDARR's reference copilot pipeline diagram.

set M) is:

$$\delta(S, G) = |M| - \max_{\phi} \sum_{i} |S_i \cap G_{\phi(i)}|$$

where ϕ ranges over all one-to-one mappings from system clusters $\{S_1, \ldots, S_m\}$ to gold clusters $\{G_1, \ldots, G_n\}$. $\max_\phi \sum_i |S_i \cap G_{\phi(i)}|$ represents the largest possible total overlap of mentions once we align each system cluster S_i to a gold cluster $G_{\phi(i)}$. The difference from $|M|$ is then the minimum number of "moves" needed.

The **CEAF**$_E$ (Luo, 2005) metric, for instance, uses a partial-similarity measure between each pair of clusters (S_i, G_j), instead of counting raw overlap. This is defined as:

$$\text{similarity}(S_i, G_j) = \frac{2|S_i \cap G_j|}{|S_i| + |G_j|}$$

Once we obtain the optimal mapping ϕ between system and gold clusters, the resulting sum is normalized, making it a single percentage-like measure.

We assessed how well these coreference resolution models, originally designed for surface text, transfer to AMR concept clustering and thereby potentially reduce the theoretical annotator workload. Specifically, we compared the copilot's automatically generated clusters to human-annotated references in terms of mention identification and the CEAF$_E$ metric. Two models, FastCoref (Otmazgin et al., 2022) and LingMess (Otmazgin et al., 2023) — were evaluated on the same dataset. Table Table 1 reports their mean precision (P), recall (R), and F1 (with standard deviations) for CEAF$_E$. The Mean F1 of CEAF$_E$ reflects overall accuracy and thus approximates the theoretical workload reduction. Meanwhile, mention identification indicates the mismatch of AMR concepts and and textual mentions used in classic coreference resolution.

In addition to our theoretical evaluation, we conducted a preliminary user study on user behavior and interaction. Two expert annotators and two non-experts were each assigned four documents to annotate using LiDARR for coreference resolution. For the first document, users received suggestions from three sources—FastCoref, LingMess, and a human annotator—and were instructed to edit existing clusters by adding or removing AMR mentions. This setup enabled measurement of alignment between user-defined clusters and model-generated ones. Human suggestions served as the performance upper bound. Table 2 shows the empirical edit distances from this study, indicating the impact of each copilot on user decisions.

	Human	LingMess	FastCoref
User1	1	6	**8**
User2	3	8	**9**
User3	2	9	**10**
User4	4	**15**	12

Table 2: Comparison of Edit Distance for Human, LingMess, and FastCoref as Coreference Suggestion Providers

Each user completed full coreference annotation tasks on three remaining documents using three different copilot interface designs, with only human-generated suggestions provided. Users then ranked the interfaces by preference. Both experts rated the interactive helper highest, followed by building from scratch, and the static helper last. Among non-experts, the interactive and static helpers were tied, with building from scratch ranked lowest. Preference scores (3 points for highest, 2 for middle, 1 for lowest) were: interactive (11), static (7), and no helper (6). We also evaluated system response time for alignment and coreference models on a server

	CEAF$_E$						mention identification					
	Prec.(%)		Rec.(%)		F1(%)		Prec.(%)		Rec.(%)		F1(%)	
Model	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
FastCoref	37.23	12.38	40.37	14.67	37.78	11.21	88.92	15.57	88.42	13.35	86.22	6.99
LingMess	**43.66**	8.39	**46.08**	12.42	**44.22**	8.99	**90.13**	13.50	**95.13**	13.23	**91.44**	10.04

Table 1: AMR coreference resolution performance with LiDARR's pipeline suggestions

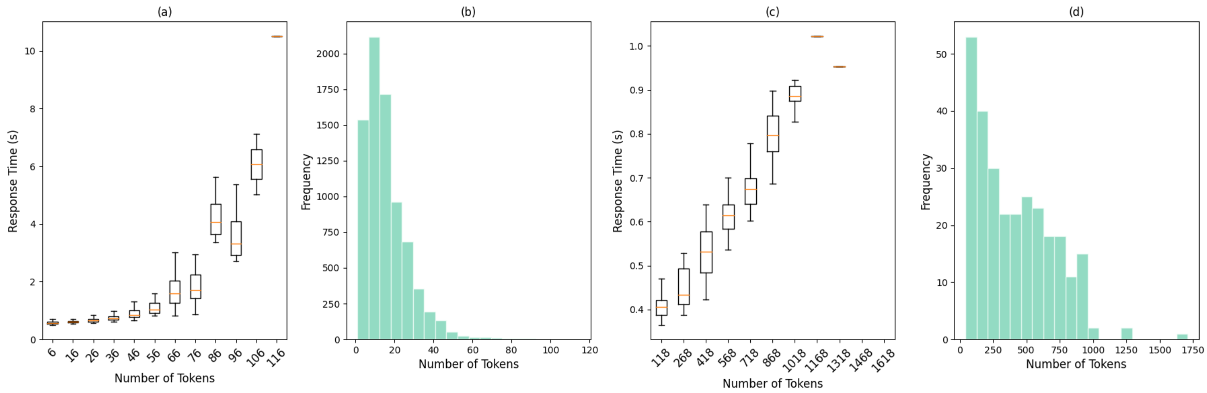


Figure 5: Response time statistics for the training set of MS-AMR: (a) Box plot of the alignment copilot's response time relative to input length (number of tokens); (b) Distribution of sentence lengths; (c) Box plot of the coreference resolver copilot's response time relative to document length (number of tokens); (d) Distribution of document lengths. In each box plot, the red line indicates the median, the box represents the inter-quartile range (IQR), and the whiskers extend to 1.5 times the IQR. Two outlier documents exceeding 2,000 tokens were excluded from analysis.

with a 24-core Intel Xeon CPU and two NVIDIA Titan Xp GPUs, one per model. The LingMess coreference model completed clustering in under one second, even for longer documents. The alignment copilot accounted for most latency, though its runtime remains acceptable if integrated during sentence-level AMR annotation. Detailed results are shown in Figure 5.

5 Conclusion and Future Work

LiDARR leverages model assistance to streamline deep semantic annotation yet UI design still shows a significant impact for user experience. Powerful AI tools need human-centered design to collaborate effectively.

An immediate downstream application following the acquisition of gold-standard annotation is the development of a knowledge graph system. This system can verify the validity of the information encoded within the semantic network. Proper visualization of the annotated semantic network is another planned area of future work, particularly since our research aims to provide verifiable knowledge support to students in classroom settings.

We are exploring UI/UX designs to unify the interfaces for bridging relations and identical coreference clusters, given their structural similarity, while minimizing potential user confusion. The interface will be refined based on further user feedback.

In brief, LiDARR is an advancement in semantic annotation tooling, combining AI-driven support with user-centric design. As development progresses, we expect LiDARR to become a valuable tool for computational linguistics and AI research.

Limitations

LiDARR's annotation logic is based on a set of assumptions widely accepted by the NLP community regarding the task formulation of coreference resolution. However, the foundational elements of this task are not without contention. There are ongoing debates in linguistics and language philosophy about what constitutes valid discourse entities for coreference tasks.

Natural language supports discourse deixis, where anaphora refers to entire discourse segments—often beyond LiDARR's coreference and alignment model. Designed solely for English, it may overlook language-specific nuances. LiDARR focuses on sub-graph alignment between AMRs and texts, yet some semantics remain encoded in AMR edges, limiting granularity. Lastly, LLM-based interpretation may pose privacy concerns, but LiDARR can work with private LLMs if needed. User discretion is advised with respect to this feature.

Ethics Statement

LiDARR aims to enhance human-computer interaction through thoughtful UI design and model assistance. A key ethical consideration is ensuring that our annotators understand how the suggestion models operate and their aforementioned limitations. We commit to providing transparent documentation and a user manual. Moreover, user privacy and copyright are of great importance to us. No documentation data will ever be collected without explicit consent, respecting both user privacy and intellectual property rights.

In addition, we are committed to fairness and reducing bias by regularly evaluating the models and incorporating diverse datasets to ensure broad applicability. We also prioritize transparency by explaining model suggestions and communicating system limitations.

Acknowledgement

This research was funded by the NSF National AI Institute for Student-AI Teaming (iSAT) under grant DRL 2019805. All views presented here are solely those of the authors and do not reflect the stance of the NSF. We are sincerely grateful to Adam Zheng for his assistance with the technical aspects of hosting. We also acknowledge the valuable computational support provided by Chameleon Cloud (Keahey et al., 2020), which was instrumental in enabling this research.

References

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.

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.

Austin Blodgett and Nathan Schneider. 2021. Probabilistic, structure-aware algorithms for improved variety, accuracy, and coverage of AMR alignments. 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 3310–3321, Online. 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.

Wei-Te Chen and Will Styler. 2013. Anafora: A web-based general purpose annotation tool. In *Proceedings of the 2013 NAACL HLT Demonstration Session*, pages 14–19, Atlanta, Georgia. Association for Computational Linguistics.

Richard Eckart de Castilho, Éva Mújdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. 2016. A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84, Osaka, Japan. The COLING 2016 Organizing Committee.

Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. 2018. Cypher: An evolving query language for property graphs. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, page 1433–1445, New York, NY, USA. Association for Computing Machinery.

Michael Wayne Goodman. 2019. AMR normalization for fairer evaluation. In *Proceedings of the 33rd Pacific Asia Conference on Language, Information, and Computation*, pages 47–56, Hakodate.

Michael Wayne Goodman. 2020. Penman: An open-source library and tool for AMR graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 312–319, Online. Association for Computational Linguistics.

Kate Keahey, Jason Anderson, Zhuo Zhen, Pierre Riteau, Paul Ruth, Dan Stanzione, Mert Cevik, Jacob Colleran, Haryadi S. Gunawi, Cody Hammock, Joe Mambretti, Alexander Barnes, François Halbach, Alex Rocha, and Joe Stubbs. 2020. Lessons learned from the chameleon testbed. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC '20)*. USENIX Association.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. AMR beyond the sentence: the multi-sentence AMR corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2022. F-coref: Fast, accurate and easy to use coreference resolution. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 48–56, Taipei, Taiwan. Association for Computational Linguistics.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2023. LingMess: Linguistically informed multi expert scorers for coreference resolution. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2752–2760, Dubrovnik, Croatia. Association for Computational Linguistics.

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

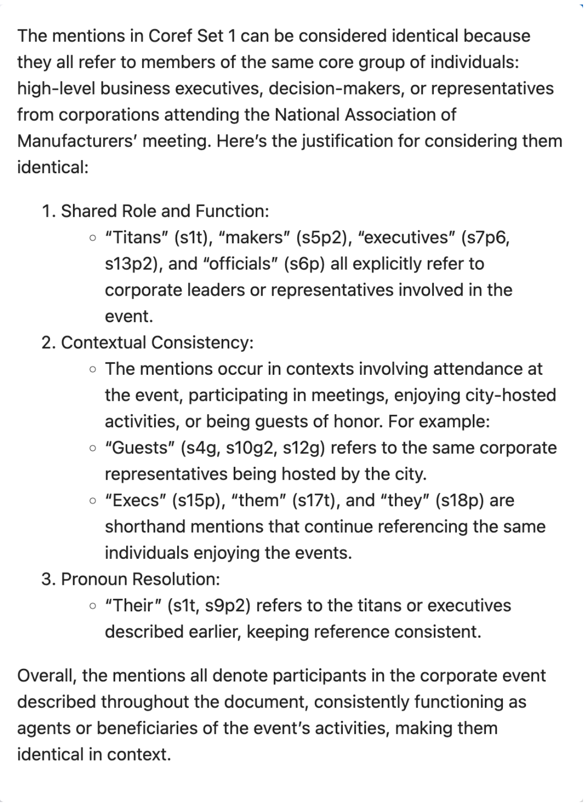
Conger, Tim O'gorman, James Gung, Kristin Wright-bettner, and Martha Palmer. 2022. PropBank comes of Age—Larger, smarter, and more diverse. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 278–288, Seattle, Washington. Association for Computational Linguistics.

Eric Prud'hommeaux and Andy Seaborne. 2008. Sparql query language for rdf. W3C Recommendation. Accessed: 2008-01-15.

Jin Zhao, Nianwen Xue, Jens Van Gysel, and Jinho D. Choi. 2021. UMR-writer: A web application for annotating uniform meaning representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 160–167, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Cluster Interpreter

This interpreter mode can be turned on via Interpreter switch in the toolbar. Figure 6 illustrates an example response from GPT-4o (OpenAI et al., 2024), interpreting the "executives" cluster and offering insights into its inferred meaning and contextual role. The prompt we used to generate the example response above is: *Given the following document and its corresponding AMRs, please provide your best justification for why the mentions listed below should be considered identical entities. <Start Document> {Document} <End Document> <Start AMRs> {AMRs} <End AMRs> <Start coref Set> {AMR coreference mentions} <End coref set> Please briefly explain how these mentions can be interpreted as referring to the same entity. Thank you!*



The mentions in Coref Set 1 can be considered identical because they all refer to members of the same core group of individuals: high-level business executives, decision-makers, or representatives from corporations attending the National Association of Manufacturers' meeting. Here's the justification for considering them identical:

1. Shared Role and Function:
    - "Titans" (s1t), "makers" (s5p2), "executives" (s7p6, s13p2), and "officials" (s6p) all explicitly refer to corporate leaders or representatives involved in the event.
2. Contextual Consistency:
    - The mentions occur in contexts involving attendance at the event, participating in meetings, enjoying city-hosted activities, or being guests of honor. For example:
    - "Guests" (s4g, s10g2, s12g) refers to the same corporate representatives being hosted by the city.
    - "Execs" (s15p), "them" (s17t), and "they" (s18p) are shorthand mentions that continue referencing the same individuals enjoying the events.
3. Pronoun Resolution:
    - "Their" (s1t, s9p2) refers to the titans or executives described earlier, keeping reference consistent.

Overall, the mentions all denote participants in the corporate event described throughout the document, consistently functioning as agents or beneficiaries of the event's activities, making them identical in context.

Figure 6: An interpretation over the cluster "executives" from GPT-4o; Example document is from AMR 3.0 multisentence AMR corpus