GLAMR: Augmenting AMR with GL-VerbNet Event Structure

Jingxuan Tu[1], Timothy Obiso[1], Bingyang Ye[1], Kyeongmin Rim[1]
Keer Xu[1], Liulu Yue[1], Susan Windisch Brown[2], Martha Palmer[2], James Pustejovsky[1]
[1]Brandeis University, Waltham, MA, USA
[2]University of Colorado Boulder, Boulder, CO, USA
[1]{jxtu, timothyobiso, byye, krim, keerxu, liuluyue, jamesp}@brandeis.edu
[2]{susan.brown, martha.palmer}@colorado.edu

Abstract

This paper introduces GLAMR, an Abstract Meaning Representation (AMR) interpretation of Generative Lexicon (GL) semantic components. It includes a structured subeventual interpretation of linguistic predicates, and encoding of the opposition structure of property changes of event arguments. Both of these features are recently encoded in VerbNet (VN), and form the scaffolding for the semantic form associated with VN frame files. We develop a new syntax, concepts, and roles for subevent structure based on VN for connecting subevents to atomic predicates. Our proposed extension is compatible with current AMR specification. We also present an approach to automatically augment AMR graphs by inserting subevent structure of the predicates and identifying the subevent arguments from the semantic roles. A pilot annotation of GLAMR graphs of 65 documents (486 sentences), based on procedural texts as a source, is presented as a public dataset. The annotation includes subevents, argument property change, and document-level anaphoric links. Finally, we provide baseline models for converting text to GLAMR and vice versa, along with the application of GLAMR for generating enriched paraphrases with details on subevent transformation and arguments that are not present in the surface form of the texts.

Keywords: AMR, VerbNet, Lexical semantic resource, Event annotation, Generative Lexicon

1. Introduction

Abstract Meaning Representation (AMR) (Banarescu et al., 2013) is a general-purpose semantic encoding for language that has become popular for its simple structure, ease of annotation, and available corpora, and whose emphasis on predicate-argument structure has proved effective for many NLP tasks (Lim et al., 2020; Zhang et al., 2021; Huang et al., 2022; Yang et al., 2023). By focusing on PropBank verb senses (Palmer et al., 2005) as the root node of the graph, the default semantic interpretation of an AMR is either: (i) a proposition with an atomic predicate and its arguments; or (ii) a Davidsonian event and its participants (Parsons, 1990). This convention, however, makes it difficult to express the richness of event structure inherent in natural language predicates, impacting any subsequent logical inference requiring reasoning over changes to argument properties or the dynamics inherent in a textual narrative.

The most recent release of the lexical resource VerbNet (VN) (Brown et al., 2019, 2022) includes predicate frame files encoded with rich subeventual semantics (Pustejovsky, 1995), modeled on Generative Lexicon (GL) Dynamic Event Structure (Pustejovsky and Moszkowicz, 2011; Jezek and Pustejovsky, 2019). In this paper, we develop an extension to AMR which incorporates the GL-VerbNet (GL-VN) enhancements in event structure, called Generative Lexicon AMR (GLAMR). The resulting meaning representation captures two essential innovations from GL: subevent structure for predicates; and designated opposition structure for the argument(s) undergoing change. This can be seen as extending the existing benefits accompanying VN frame file specifications: sense-specific syntactic construction files; semantic role designations for arguments; and sense clustering (Schuler, 2005; Kazeminejad et al., 2022; Stowe et al., 2021). We illustrate a generic graph representation of GLAMR in Figure 1. The GL event graph is attached to the root node of the AMR graph through the *event structure* edge. GLAMR also includes a separate graph that contains document-level anaphoric relations between entities.

Figure 2 depicts the main part of the graph structure of the GLAMR for the sentence *Slice the onions*. We extend the classic AMR graph with the GL event structure through the role `:event-structure`. This maintains the flexibility and portability of GLAMR for further extensions in the future. The role `SubEvents` is the root node that governs all the subevents denoted as `E1`, `E2`, etc. Roles from the AMR will be re-entrant to the GL event subgraph when they are mentioned again (e.g., `o/onions` as a `PATIENT` from `E1`). We leave the description of the document-level anaphora graph in the later sections.

We outline the major contributions of this paper as follows: (1) we introduce GLAMR, a new semantic representation extending AMR with GL event structure; (2) we propose a pipeline for automatic augmentation of AMR to GLAMR graphs;

LREC-COLING 2024, pages 7746–7759

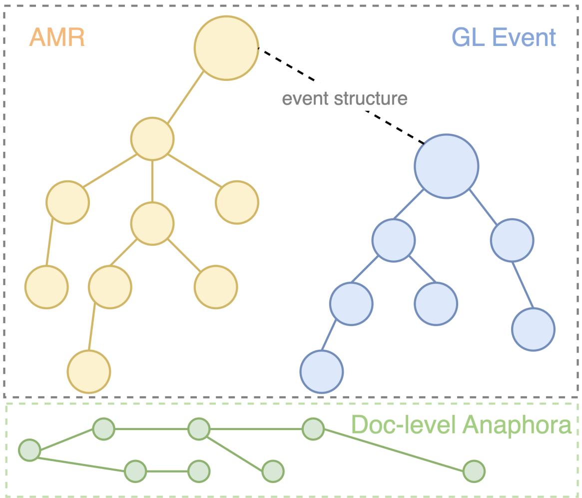

Figure 1: Generic Representation of GLAMR graph.

(3) we create a GLAMR dataset from procedural texts by producing gold AMRs and converting them into GLAMR with the proposed pipeline; (4) we explore baseline models for converting text to GLAMR and GLAMR to text, describing a sound setup for generating enriched paraphrases with details on subevent transformations and implicit objects mentions. We make our code and data publicly available.[1]

2. Related Work

Semantic meaning representations have become popular and useful to various natural language understanding tasks that can benefit from encoding the meaning of the texts in a structured way. To this end, several different meaning representation formalisms have been developed for NLP tasks, such as Minimal Recursion Semantics (MRS) (Copestake et al., 2005), Universal Conceptual Cognitive Annotation (UCCA) (Abend and Rappoport, 2013), Universal Decompositional Semantics (UDS) (White et al., 2016), AMR (Banarescu et al., 2013), Universal Meaning Representation (UMR) (Van Gysel et al., 2021) and BabelNet Meaning Representation (BMR) (Lorenzo et al., 2022).

Of these languages, AMR is currently the most widely adopted meaning representation, and there are several previous papers to extend AMR with additional semantic information for specific NLP tasks. O'Gorman et al. (2018a) and (Naseem et al., 2022) created document-level AMR by linking cross-sentential entities through coreference relations. Bonial et al. (2020) extended AMR to dialogue systems by incorporating cues such as *how* and *what* on what is said, within a dialogue act annotation specification. AMR has also been applied to represent spatial information (Bonn et al., 2020), gestures (Brutti et al., 2022) and actions (Stein et al., 2023; Tam et al., 2023). Our work, in the spirit of AMR extensions, is the first to propose a representation that incorporates GL event structure into AMR.

VN is a resource for English that derives from Levin's verb classification (Levin, 1993). Its goal is to provide semantic representations for a wide coverage of verb classes (Dang et al., 1998; Schuler et al., 2000; Schuler, 2005; Schuler et al., 2008) and rich event structures (Brown et al., 2019, 2022). It has been leveraged to improve various NLP tasks such as semantic role labeling (Giuglea and Moschitti, 2006) and word sense disambiguation (Brown et al., 2011). In addition, VN has also been used for event understanding tasks such as event tracking (Dalvi et al., 2019), video event understanding (Monfort et al., 2018) and story generation (Ammanabrolu et al., 2019). In this work, we take advantage of the semantic representations from both VN and AMR, and generate GLAMR as a unified representation that we hope can be useful for future downstream tasks.

AMR parsing (text-to-graph) and AMR-to-text generation are the two benchmarking tasks for AMR. Early parsing approaches are mainly statistics-based and transition-based (Flanigan et al., 2014, 2016; Ballesteros and Al-Onaizan, 2017; Liu et al., 2018). With the advent of large pretrained language models, more recent approaches train transformer models on both text and graph data to integrate linearized and structural information in the model training (Cai and Lam, 2020; Bevilacqua et al., 2021; Bai et al., 2022). In this work, we fine-tune the pretrained BART models (Lewis et al., 2020) from Bai et al. (2022) and evaluate the results. These results establish the first baseline for GLAMR parsing.

3. Building Blocks

To incorporate subevent structure into AMR, we employ the Coreference under Transformation Labeling (CUTL) dataset (Rim et al., 2023) for the GLAMR annotation, and we use the lexical resource GL-VN (Brown et al., 2019, 2022) to retrieve event semantics.

CUTL is a dataset with annotations of entities, their anaphoric and coreference relations, and the accompanying event semantics over the R2VQ dataset of cooking recipes (Tu et al., 2022a). It annotates each transformation event as an I/O process where the explicit and implicit arguments (en-

[1] https://github.com/brandeis-llc/GLAMR-LREC-COLING-2024

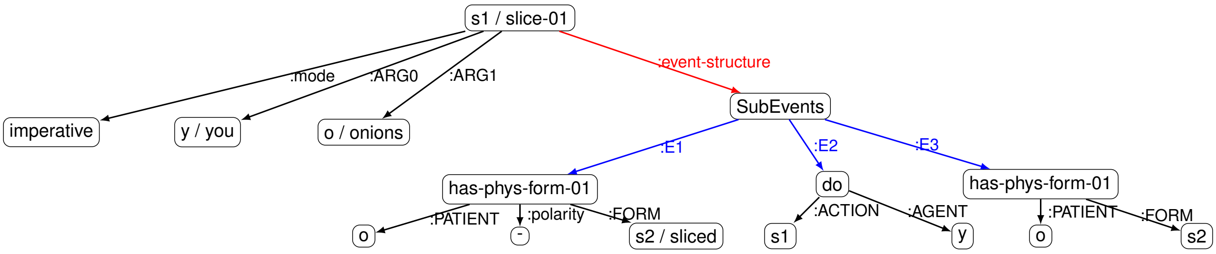

Figure 2: GLAMR graph of the sentence *Slice the onions.*

tities) annotated as input or output, as well as different anaphoric relations between the entities including *Coreference under Identity (CuI), Coreference under Transformation (CuT), Change of Location (CoL), Aggregation, Separation*. These annotations in procedural texts are particularly useful to our task because they tend to be task-oriented and highly contextualized, reflecting the richer representation of the subevents that are taking place in the course of a sequence of events in the narrative.

GL-VerbNet has recently been updated with representations for the GL event structure. The VN representation of each verb class includes its semantic roles and syntactic patterns. For each syntactic pattern, GL-VN now defines an event structure that contains subevents for tracing the property change of the arguments involved in the event. As demonstrated in Figure 3, each VN class is associated with multiple syntax frames on the possible compositions of the verb sense (left). For each frame, it shows the semantic roles and GL event structure (right).

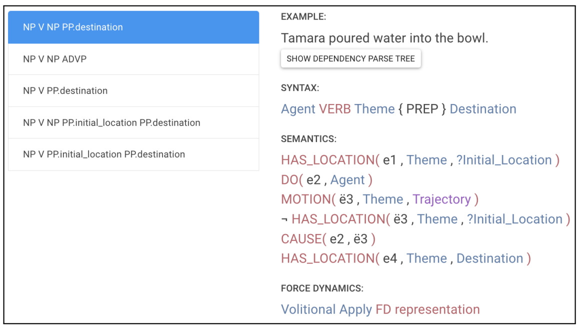

Figure 3: GL event structure of the VN class `pour-9.5`.[2]

4. Mapping from GL-VN to GLAMR

In this section, we outline the graph structure of subevents proposed in our study. We describe cases for each feature from the GLAMR graphs.

[2] https://uvi.colorado.edu/verbnet/pour-9.5

We use GL-VerbNet for identifying the canonical GL event structure of predicates and use the CUTL annotation to identify subevent role values and anaphoric relations.

Formally, given a predicate node, we introduce a new relation `:event-structure` that links the root role `subevents` as the direct child of the predicate. Thus the GL event structure in GLAMR is a portable subgraph that can be added to or detached from classic AMR graphs. A full GLAMR example with `pour-9.5` is shown in (1).

(1) ***Pour*** *them into the bowl.*

```
(p / pour-01
  :ARG0 (y / you)
  :ARG1 (t / them)
  :ARG3 (b / bowl)
  :event-structure (s / subevents
    :E0 † (d / do
      :ACTION p)
    :E1 § (h / has_location
      :THEME t ‡
      :INITIAL_LOC ‡ N/A)
    :E3 (a / and
      :op1 ¶ (m / motion
        :THEME t
        :TRAJECTORY N/A)
      :op2 (h / has_location
        :polarity - ¶
        :THEME t
        :INITIAL_LOC N/A))
    :E4 (h1 / has_location
      :THEME t
      :DESTINATION b)
  :mode imperative )
```

§ **Subevent Sequence** New AMR roles `E1`, `E2`, . . . are added to represent the subevent indices. They are aligned with the GL event structure of the predicate encoded in VN. Note that the classic GL event structure of a predicate contains many subevents. However, in the GLAMR graph, we *only* include subevents that contain at least one `Patient` or `Theme` role, as this work focuses on representing argument property changes as consequence event transformation. Each selected subevent is linked to the predicate node based on the subevent index.

‡ **Subevent arguments** The concepts and variables inside the subevent will be synced with the outside through reentrance. For example, `(b / bowl)` from `:ARG3` also has the role of `:DESTINATION` in `:E4`. Missing or hidden subevent role values from the sentence context are represented as `N/A`.

† **`ACTION` Subevent** Given the nature of the procedural texts in our data, we also incorporate the `action` subevent into GLAMR to represent the action that has been performed on the objects during the event time. Some GL-VN verbs have a `do` subevent. We make it universal in GLAMR and distinguish it by fulfilling the subevent with the `:ACTION` from the predicate verb. The concept of the role `:ACTION` is represented as the verb lemma of the predicate.

¶ **Simultaneous Subevents and Negation** GL event structures contain subevents that may occur simultaneously, denoted by temporal Allen relations (Pustejovsky, 1995; Allen, 1983). In GLAMR, subevents with the same temporal index are stacked with the `:op` roles. GL event structure also uses standard logical notation, "¬" to represent the negation of the subevent predicate. For example, ¬`COOK` means the subevent "uncooked" or "not cooked". In GLAMR, we use the attribute `:polarity` to represent negation.

Implicit Objects The subevents from GL-VN include all the dominant roles that are associated with the predicate. However, not all the arguments involved in the subevents are explicitly mentioned in the texts. To fill in the missing information, we introduce new roles in GLAMR:

(2) ***Slice*** *bananas in half.*

```
(s1s / slice-01
  :ARG1 (s1b / bananas)
  :manner (s1h / half)
  :event-structure (s1se1 / subevents
    ...
  :imp_output (s1s1 / RES.slice))
  :mode imperative)
```

Fry *in the butter until soft.*

```
(s2f / fry-01
  :medium (s2b / butter)
  :event-structure (s2se / subevents
    :E1 (s2c / cooked
      :polarity -
      :IMP_PATIENT (s2s1 / RES.slice)
      :V_FINAL_STATE (s2f1 / fried)))
    ...
  :imp_output (s2f / RES.fry)
  :mode imperative)
```

In the subevent structure, we prepend `IMP_` to the `PATIENT` or `THEME` role to indicate it is implicit. Following the CUTL annotation, the role value is represented as `RES.[event]`. In the first graph above, the implicit patient `RES.slice` is the result of the event `slice`. Outside of the event structure, we also introduce `:imp_output` for implicit output from the current event. These implicit objects provide richer context for the events, as well as enable the anaphoric relations under the document setting.

Document-Level Anaphora We generate document-level GLAMR (DocGLAMR) to express anaphoric relations between entities. Following the UMR guideline[3], we generate a DocGLAMR as a separate graph for each sentence-level GLAMR.[4] We borrowed the UMR `:same-entity` role to accommodate the richer anaphoric relations from the CUTL data in DocGLAMR.

(3) ***Fry*** *in butter until soft.* (document-level)

```
(s2 / sentence
    :coref ((s1s1 :cui s2s1)
            (s2s1 :aggregation s2f)
            (s2b :aggregation s2f))
```

(3) is the DocGLAMR graph for the second sentence from example 2. Anaphoric relations in the DocGLAMR graph may include roles from the graphs of current and earlier sentences. For example, `s1s1/RES.slice` from the first sentence and `s2s1/RES.slice` from the second sentence are coreference (`:cui`). Both `s2s1/RES.slice` and `s2b/butter` are aggregated into `s2f/RES.fry` through the *fry* event (`:aggregation`).

5. GLAMR on Procedural Texts: Annotation Task

In this section, we report on the methodology and results of the GLAMR annotation experiment on the procedural texts. We present the approach to inserting the GL event structure of the predicate into the AMR graph and identifying the subevent role values from the semantic roles. We build our dataset on top of the existing CUTL corpus (Rim et al., 2023) as it already contains entity annotation (explicit or implicit event input and output), Semantic Role Labeling (SRL) annotation

[3] https://github.com/umr4nlp/umr-guidelines/blob/master/guidelines.md

[4] MSAMR (O'Gorman et al., 2018b) and DocAMR (Naseem et al., 2022) also proposed document-level AMRs. They present a unified graph with embedded coreference chains. We use the UMR specification as it fits better to our need to keep the subevent subgraphs detachable from class AMR graph. In addition, UMR can also be mapped to AMR easily (Bonn et al., 2023).

	R1 (15)	R2 (25)	R3 (25)	Overall (65)
SMATCH	88.1	85.4	92.5	88.8
SemBleu	77.5	69.1	83.3	76.5
AnCast	82.9	79.8	88.0	83.6

Table 1: Number of recipes and IAA of AMR annotation from each round (R).

(predicate-argument structure from the VerbAtlas frames and arguments (Di Fabio et al., 2019)), and coreference annotation (document-level coreference graph from the entities).

5.1. AMR Preparation

To create our dataset, we first randomly sample 65 recipes from the CUTL dataset. Each recipe contains the raw text and annotation of 5-10 sentences. To create the gold AMR graphs for the recipes, we first adopt the recent AMR parser (Drozdov et al., 2022) to parse each recipe sentence into a PENMAN graph, and ask annotators to annotate and validate the parser output. Annotation of the AMR graphs was done in 3 rounds by 5 researchers and graduate students from the linguistics and computer science departments of a US-based university. All annotations were conducted using UMR Writer, an annotation tool for editing and generating graph-based meaning representations (Zhao et al., 2021). Each graph is dually annotated and Inter-Annnotator Agreement (IAA) is computed at the end of each round. Pairs of annotators then meet to adjudicate disagreements and create a finalized gold standard AMR annotation.

Table 1 shows the IAA scores from the annotation. We use SMATCH (Cai and Knight, 2013), SemBleu (Song and Gildea, 2019) and Labeled relation match score of Ancast metrics (AnCast) (Sun and Xue, 2024) as our primary IAA metrics, which are uniformly high across different rounds with a mean SMATCH of 88.8, SemBleu of 76.5 and AnCast of 83.55. Overall, we produce a dataset of 486 gold AMR graphs with 748 predicates from 65 cooking recipes. We demonstrate the process to transform them into GLAMR graphs in the following sections.

5.2. GL Event Structure Identification

5.2.1. VN Class Mapping

To identify the GL event structure for GLAMR, we start by mapping the PropBank (PB) sense of the AMR predicate to the VN class, which is the entry to frames and subevents that are associated with that verb class. Specifically, we utilize the files in SemLink (Stowe et al., 2021) to establish a mapping between the PB verb sense and the corresponding VN class. For the PB verb sense with multiple mappings, we select the most accurate VN class from the candidates (e.g., PB sense `chop.01` can be linked to `carve-21.2-2` or `cut-21.1-1` in VN). For a predicate with no curated VN class, we link it to its semantically closest VN class (e.g., predicate *debeard* is close to `remove-10.1` in VN).

5.2.2. VN Frame and Subevent Identification

Considering that each VN class can have multiple different frames consisting of a semantic representation partnered with a subevent structure (Schuler, 2005) (Example in Figure 3), we take advantage of the SRL annotation from the CUTL dataset to identify the closest matching VN frame and its unique subevent structure for each predicate. An example sentence (a.) with its SRL (b.) and matching VN frame semantics (c.) is shown in (4) below.

(4) a. *Remove[remove-10.1] the onions to a dish.*
b. `V Patient Destination`
c. `Agent V Theme {PREP} Init_Location`

Given the nature of procedural texts that involve rich action-related arguments (*location, instrument, etc*) and implicit objects (*agent, patient, etc*), it is not always guaranteed we can identify the frame that matches exactly to the sentence semantic roles as shown in Example 4. Thus we propose a heuristics-based pipeline that enables a fuzzy match between the sentence SRL and VN frame candidates to identify the most precise frame and its corresponding subevent structure. We succinctly summarize the heuristics as follows:

AGENT Removal Due to the imperative nature of procedural texts, the `Agent` roles are always hidden from the surface form, while the generalized form of the VN frame always starts with the `Agent`, which is excluded for the frame matching.

Hidden Role Insertion While being a key anchor in determining the matching frame, the `Patient/Theme` role is sometimes hidden in the recipe text. When this role is missing from the sentence SRL, we insert a `Patient` immediately following the `V` role.

Interchangeable Roles Distinguishing between `Patient` and `Theme` can be challenging, leading to discrepancies between SRL annotation and VN frames. Following the VN role hierarchy, we overlook the nuanced distinctions between the two roles (all are subtypes of `Undergoer`), and treat them as interchangeable for matching. The same change also applies to the set of `Location` and `Destination`; and the set of `Product`, `Result`, and `Goal`.

Heuristic	Coverage (%)
Remove AGENT	14.8
+ Insert hidden role	18.7
+ Interchange roles	71.2
+ Reverse role order for light verb	73.4
+ Flexible role order	96.8
+ Max role match (default)	100.0

Table 2: VN frame matching coverage of the 748 predicates from the 65 recipes.

Light Verb Construction Reversal Most VN frames are constructed following a general syntactic structure where the verb proceeds the objects. To match against them, we reverse the order of the `Patient/Theme` and the `V` role when a light verb appears. For example, the sentence *let meat **cook*** is converted to ***cook** meat*; and *bring to a **boil*** to ***boil** (water)*.

Flexible Role Order The syntactic structure of the sentence can vary while conveying the same semantic meaning. For example, the position of the `Location` role can vary significantly in a sentence in our data (e.g., ***In a large skillet** melt butter . . .*), while it is typically positioned at the sentence end in VN frames. To match those frames, we only limit the relative order of the `Patient/Theme` and the `V`, all remaining roles such as `Location` and `Instrument` are treated as optional, and their orders can be flexible.

We briefly describe the process of the identification pipeline. First we remove the agent and insert hidden roles before the first match checking. Then we apply the other heuristics in order until we find a match. If there is still no matching frame, we apply a default rule that selects the frame with the most matching roles with the sentence SRL. Table 2 shows how the matching coverage increases when adding the heuristics sequentially. Due to the syntactic variability of the sentences from the data, the coverage rate is relatively low (18%) with only the first two heuristics. *Interchange roles* is the most effective heuristic in the pipeline that increases the coverage to 71%. Overall it shows that the heuristics greatly enhance the coverage rate in the identification of matching VN frame.

Given the matched VN frame and its subevent, we fulfill the subevent role values by identifying the text spans with the same semantic role in the SRL annotation. We also adopt the *interchange roles* heuristics to mitigate the confusion between similar roles such as `Patient` and `Theme`. For any subevent role that involves the state of the predicate verb[5], we fulfill its value with the past participle form of the predicate verb (e.g., *fried* in (2)) to indicate the property change of the event (Tu et al., 2022b). For subevent roles that are hidden from the text, we fulfill them with the CUTL I/O annotation.

[5]In VN, it is denoted by the role name that starts with `V_*` such as `V_State`, `V_Form`, etc.

Sube. Names	Count	Sube. Roles	Count
`[¬]has_loc.`	636 (26%)	`Patient`	2038 (38%)
`[¬]cooked`	286 (11%)	`Theme`	1131 (21%)
`[¬]MI_state`	274 (11%)	`V_Final_State`	378 (7%)
`[¬]together`	212 (9%)	`Initial_Loc.`	371 (7%)
`motion`	211 (9%)	`V_State`	274 (5%)

Table 3: Most frequent subevent names and roles from the GLAMR data. `MI_state` stands for `has_material_integrity_state`.

We also evaluate our heuristics pipeline by manually checking the correctness of the matched frames of 100 predicates randomly sampled from the whole dataset. In the end, the matching accuracy is 96% on the sample set, suggesting the robustness of our pipeline. The errors are mainly from the wrong VN sense or special frame syntax that is not captured by the heuristics.[6] We fix the annotation errors accordingly and will evaluate on the full set of predicates in the release of the data.

5.3. GLAMR Generation

We generate the final GLAMR graphs by integrating the fulfilled subevent structure into the gold-standard AMR graphs. Operationally, we use the token alignment to identify the position of the predicate in AMR and the text spans of the semantic roles to be inserted. We adopt the open-source library `Penman` (Goodman, 2020) to generate and edit the GLAMR graphs based on the specifications defined in §4.

We list the frequent subevent names and roles from our data in Table 3 to show the enrichment from the GL-VN in GLAMR. For the subevent names, most of them involve the change of locations (`[¬]has_loc.`) that is triggered by predicates such as *place, remove, add, etc*. `[¬]cooked` and `[¬]MI_state` involve the state or physical change of the objects triggered by predicates such as *bake, cut, saute, etc*.

Correspondingly, the most frequent subevent roles involve objects that undergo the change (`Patient/Theme`). Other frequent roles such as locations (`Initial_Loc.`) and argument state from the event (`V_Final_State/V_State`) also provide contextualized information for the whole subevent structure. While this study focuses on the data from the cooking domain, the results from the generated GLAMR graphs indicate that GL event structure is an effective resource for enriching AMR with fine-grained subevent information.

[6]For example, `spray.9.7-2` has no frames in the VN, so we correct it to `spray.9.7`.

6. Experiments

In this section, we present experiments in the tasks for AMR and GLAMR parsing, AMR-to-text generation, and GLAMR paraphrasing with GPT-4. We start by exploring parsing and generation baselines from language models on our adjudicated dataset. Furthermore, we experiment with parsing the raw text of each sentence directly into an AMR graph under a supervised learning setting. We also experiment with generating natural language paraphrases enriched with subevents and implicit entity information from the GLAMR graphs under the prompt learning setting. Overall we randomly sample 50 recipes for training and 15 for testing.

6.1. GLAMR Parsing and Generation

We use the BART-based models (Lewis et al., 2020) as the baseline for our task. Bai et al. (2022) releases AMRBART, a set of models that are pretrained on various AMR tasks. We first finetune the AMRBART models on the training set of the GLAMR dataset for the parsing task, and evaluate the results on the test set using the SMATCH (Cai and Knight, 2013) score. We preprocess the data to obtain a linearized graph for every GLAMR graph and finetune the model for 30 epochs. For comparison, we also finetune the AMRBART models under the same settings for parsing our recipes' AMR data.

Table 4 shows the parsing results. Overall the large model performs better than the base one on both AMR and GLAMR parsing (5 and 12 SMATCH up, respectively). Compared to AMR, GLAMR parsing is more challenging to the model (22 SMATCH lower than on the large model) due to the new customized concepts and roles in GLAMR for representing GL events.

The large model also performs better than the base model on parsing the new GL event subgraph. These scores are obtained by only considering the relations `:event-structure` and `:imp_ouput`. Any other nodes or relations not necessary to fully express the GLAMR relations were removed from the AMR graphs. SMATCH is then computed with these new graphs. The large model performs better than the base model by 10.0 percentage points.

By examining the data, GLAMR fine-tuning involves 97 unique roles, while only 50 of which are found in AMR and 47 of which are only seen during the fine-tuning phase. Due to the additional roles that GLAMR introduces, a larger corpus with more examples of this complexity or external lexical knowledge could see improved results.

We summarize the common parsing errors from the models below. To avoid the overuse of the `:mod` role, in the annotation guideline, we distinguish different modifiers by indicating `:purpose`

	SMATCH (base)	**SMATCH** (large)	**BLEU** (large)
AMR	81.3	86.8	55.5
GLAMR	52.9	64.9	N/A
GL event graph	47.0	57.0	N/A

Table 4: Parsing (SMATCH) and generation (BLEU) results from the AMRBART models.

for phrases like *cooking oil*, `:consist-of` for phrases like *beef stew*, and `:source` for ingredients like *lime juice*. While this change does not contribute to any deterioration of intelligibility or fluency, the model is not able to pick up on this subtle nuance among those roles, resulting in some errors in the output. Another common error is the misplacement of some triples. For example, in the sentence *Brown the chicken on each side*, the model added the `each side` as `:location` to the predicate `brown`, while it should be added to the concept `chicken`. The model also predicts the same roles with different names such as `:imp_output`, `:imp_outputI`, and `:imp_outputJ`. While these names refer to the same role and can be normalized in postprocessing, SMATCH would not consider these as a match, despite a human believing otherwise.

We finetune another model from AMRBART (Bai et al., 2022) that is pretrained for AMR-to-text generation. We only conduct the text generation experiment from the recipe AMR, as there is no gold-aligned text available mainly for the subevents from GLAMR. We use the BLEU score (Papineni et al., 2002) as the evaluation metric to compare the generated text to the original recipe text. Table 4 shows the generation results with the BLEU score of 55.5, which indicates large amounts of information are being preserved in the model output (Lavie, 2011).

We summarize the common generation errors. The model confuses singular and plural nouns. For example, it generates *pour into bowls* instead of *bowl*. Our recipes also contain information in parentheses (e.g. *Serve pork with rice (if desired)*). When generating the text from the AMR graph, parentheses are never used, affecting the linearity of the parenthetical and the surrounding words. Another error is preposition choice (e.g. *with*/*on rice*). As we are only providing one reference per translation, any deviation from identical sentences is penalized.

6.2. GPT Paraphrasing with GLAMR

Motivated by the concept of *dense paraphrasing* that is first proposed by (Tu et al., 2022b, 2023) to enrich surface texts with contextualized and implicit information, We develop experiments that generate paraphrases enriched with subevents and implicit entities from the GLAMR structures. We formalize the task as zero-shot prompting and apply GPT-4

Prompt 1	
Context	(in natural language) *Heat oil in a skillet , add chicken and brown 3 - 5 minutes per side , remove .*
NL Generation Target	(in graph form) *Add onions and garlic to pan , cook for 2 minutes .*
Response given AMR	**Response given GLAMR**
Add both onion and garlic to the pan and let them cook for a duration of two minutes.	Next, you need to add onions and garlic to the pan. When you combine these ingredients in the pan, it forms a new mixture. Then, you should cook this mixture for about 2 minutes. During the cooking process, heat is applied to the mixture, transforming it from its raw state to a cooked state. This results in a new output, which is the cooked mixture of onions and garlic.
Prompt 2	
Context	(in natural language) *Wash cauliflower well and remove the stems.*
NL Generation Target	(in graph form) *Boil until tender.*
Response given AMR	**Response given GLAMR**
Ensure that you boil the cauliflower until it becomes tender.	Following the washing and stem removal of the cauliflower, you are instructed to boil it. This boiling process should continue until the cauliflower reaches a tender state. The cauliflower, which was previously raw and then had its stems removed, is now being subjected to heat. This transition from being uncooked to being boiled is a significant change in its state. The end result of this process is boiled cauliflower, which is a combination of the previously washed and de-stemmed cauliflower.
Prompt 3	
Context	(in natural language) *N/A*
NL Generation Target	(in graph form) *Preheat oven to 350 degrees F and line a cookie sheet with parchment paper.*
Response given AMR	**Response given GLAMR**
Start by warming up the oven to a temperature of 350 degrees Fahrenheit. Following that, prepare a cookie sheet by covering it with parchment paper.	Begin by preheating your oven to a temperature of 350 degrees Fahrenheit. Following that, line your cookie sheet with parchment paper.
Prompt 4	
Context	(in natural language) *Mix in the lemon juice and cream.*
NL Generation Target	(in graph form) *Season and serve.*
Response given AMR	**Response given GLAMR**
Alter adding the lemon juice and cream, proceed to season the mixture. Once done, It's ready to be served.	After you have mixed in the lemon juice and cream, proceed to season the mixture. This action will transform the mixture into a seasoned version. Following this, serve the seasoned mixture. This will result in a change of possession from you to the recipient, creating a served dish.

Table 5: Examples of GPT-generated paraphrases from AMR and GLAMR graphs. Each example shows the target and its context (previous sentence), as well as the paraphrases from different prompts. Highlighted spans indicate the text relating to the subevents and argument property change from the GLAMR paraphrases.

(Brown et al., 2020; OpenAI, 2023) to generate the paraphrases. In this prompt, we include a human-created instruction on the task and relevant information including the GLAMR, DocGLAMR of the current sentence (to be paraphrased), and the text of the previous sentence (context for implicit objects and coreference). As a comparison, we also run GPT-4 on the prompt by replacing the GLAMR and DocGLAMR with the AMR graph. The instruction and the previous sentence context remain the same. We show the full prompts in the Appendix A.1.

We conduct the experiment on the 15 recipes from the test set and summarize our findings. The average token length of the paraphrase from the GLAMR is 67, compared to the 24 from the AMR and 14 from the raw sentence. It implies the potential richness of the paraphrase generated from the GLAMR graphs. Example paraphrases are shown in Table 5. Overall, the GLAMR paraphrases are able to recover the argument property change from the subevents as in the highlighted text spans. In Prompt 1, compared to the AMR paraphrase that only includes the full *add* and *cook* events, GPT-4 with GLAMR is able to decompose them into subevents and translate them into text paraphrases. It is also able to infer implicit subevent outputs such as *mixture* from *onions and garlic*, and *cooked mixture* from *mixture*. Prompt 2 involves an implicit argument (*cauliflower to be boiled*) that needs to be learned from previous sentences. Although both AMR and GLAMR can identify the implicit argument, GLAMR paraphrase can better express the property change of the argument by leveraging the anaphoric relations between the *cauliflower* and its *stem*. For example, it shows the transformation from *washed and de-stemmed cauliflower* to *boiled cauliflower*.

Prompt 3 involves the case when the target sentence is the first sentence of the recipe. With the lack of context and potential transformation from the event, both AMR and GLAMR paraphrases tend to just convey the meaning from the surface form of the text. Prompt 4 shows a challenging example where the implicit argument from the target sentence (*soup*) is also not explicitly mentioned in the previous sentence. While the corresponding DocGLAMR graph has long-distance anaphoric relations, GPT-4 fails to recognize it in the paraphrase. In general, the experiment shows the usefulness of the GLAMR and how it can supplement the original AMR graphs for generating enriched paraphrases or producing animations of the text with details on the event transformation and argument property change.

7. Discussion

Adjudication rules for AMR annotation We discuss the general rules established for adjudicating the AMR annotation agreements. For any predicate without proper PB sense available, we represent it as `predicate-00` in the graph (e.g. `preheat-00`), and assume basic conventional arguments for the predicate: `:ARG0` as subject; `:ARG1` as direct object; `:ARG2` as indirect object. Most sentences in procedural texts, like recipes, are imperative, so we add `:mode imperative` to the AMR graph and uniformly filled `:ARG0` with `you`. When annotating phrases like *mixing bowl*, *baking pan* and *cooking spray*, the role of gerunds are confusingly labeled as `:ARG-of`, `:mod` or `:purpose`. We agree on using `:purpose` because the gerunds specify the purpose of the instruments. We also distinguish between `:part-of` and `:source` when annotating noun-noun components. `:part-of` emphasizes the part-whole relationship between two entities (e.g., *mint leaf*) while `:source` stresses the origin of the entity (*orange juice, coconut milk*)

Frame identification with VerbNet parser We compare the VN frames identified from our pipeline and other off-the-shelf tools. VN Parser (Gung, 2020; Gung and Palmer, 2021) is a BERT-based model that can predict disambiguated VN class of the potential predicates and the VN frame instantiated with arguments extracted from the input query. First we compare the VN class identified from the VN parser and our pipeline. The parser is able to predict the VN class for 70.5% of the total 748 predicates. Out of the predicted VN classes, 371 align with our gold annotation, yielding an aggregated accuracy of 49.6%. We also compare the VN frames identified from the pipeline and the parser, producing a low agreement of 24.3%. Unlike narratives, the recipe text in our data tends to be instructional and imperative, thus making it challenging to the existing tools to parse.

8. Conclusion

We have proposed GLAMR, a meaning representation that extends AMR by incorporating the event structure proposed in the Generative Lexicon Theory. It features an interpretation of structured subevents of predicates, and opposition structure of property changes of arguments to the event. We have also created a new GLAMR dataset that consists of both sentence and document-level AMR and GLAMR graphs annotated from 486 sentences in the cooking recipe domain. We have presented a heuristic-based pipeline for converting AMR to GLAMR by leveraging the resource GL-VerbNet, and shown its robustness with high frame coverage and accuracy from the evaluation. Additionally, we have conducted experiments to build decent baselines for AMR and GLAMR parsing and discussed the model errors and challenges, which indicate the soundness of our dataset. Finally, we have explored the task for generating enriched texts from the GLAMR graphs and shown its potential usefulness for paraphrasing of the text with details on the event transformation and argument property change.

9. Limitation

The GLAMR dataset is created from the recipe text from an existing dataset, which is limited to a specific domain and prior annotations. We intend to apply our approach on data from other text genres and leverage available SRL and coreference resolution models for automatically generating the prior annotations for GLAMR. While the CUTL annotation includes implicit objects, most of those are ingredients required for the I/O process. In GLAMR, the role values of some types of subevents such as `Initial_Location` and `Instrument` cannot be automatically mapped to or recovered from the sentence context. For example, *Pour into the pot [from location]* or *Stir for 5 minutes [with instrument]*. This can be mitigated or solved by manual annotation or learning from the document-level context. In the paper, DocGLAMR is used as part of the prompt for paraphrase generation with the GPT-4. However, it is not discussed for use on parsing tasks. Following the UMR specification, DocGLAMR is a separate graph with both implicit and explicit roles reentered from the corresponding sentence-level graph. We leave it to the future discussion on the training paradigm with separate document-level graphs.

10. Acknowledgements

We would like to thank Jin Zhao for the support and instructions on GLAMR annotation with UMR Writer. This work was supported in part by NSF grant 2326985 to James Pustejovsky. The opinions and views reported herein are those of the authors alone.

11. Bibliographical References

Omri Abend and Ari Rappoport. 2013. Universal conceptual cognitive annotation (ucca). In *Annual Meeting of the Association for Computational Linguistics*.

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

Prithviraj Ammanabrolu, Ethan Tien, Wesley Cheung, Zhaochen Luo, William Ma, Lara J. Martin, and Mark O. Riedl. 2019. Story realization: Expanding plot events into sentences. *ArXiv*, abs/1909.03480.

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.

Miguel Ballesteros and Yaser Al-Onaizan. 2017. AMR parsing using stack-LSTMs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1269–1275, Copenhagen, Denmark. 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 *LAW@ACL*.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

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 Twelfth Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

Julia Bonn, Skatje Myers, Jens E. L. Van Gysel, Lukas Denk, Meagan Vigus, Jin Zhao, Andrew Cowell, William Croft, Jan Hajič, James H. Martin, Alexis Palmer, Martha Palmer, James Pustejovsky, Zdenka Urešová, Rosa Vallejos, and Nianwen Xue. 2023. Mapping AMR to UMR: Resources for adapting existing corpora for cross-lingual compatibility. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 74–95, Washington, D.C. Association for Computational Linguistics.

Julia Bonn, Martha Palmer, Zheng Cai, and Kristin Wright-Bettner. 2020. Spatial AMR: Expanded spatial annotation in the context of a grounded Minecraft corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4883–4892, Marseille, France. European Language Resources Association.

Susan Windisch Brown, Julia Bonn, James Gung, Annie Zaenen, James Pustejovsky, and Martha Palmer. 2019. Verbnet representations: Subevent semantics for transfer verbs. *ACL 2019*, page 154.

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 *International Conference on Computational Semantics*.

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

Richard Brutti, Lucia Donatelli, Kenneth Lai, and James Pustejovsky. 2022. Abstract Meaning Representation for gesture. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1576–1583, Marseille, France. European Language Resources Association.

Deng Cai and Wai Lam. 2020. AMR parsing via graph-sequence iterative inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1290–1301, Online. Association for Computational Linguistics.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Ann A. Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3:281–332.

Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, and Peter Clark. 2019. Everything happens for a reason: Discovering the purpose of actions in procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4496–4505, Hong Kong, China. Association for Computational Linguistics.

Hoa Trang Dang, Karin Kipper Schuler, Martha Palmer, and Joseph Rosenzweig. 1998. Investigating regular sense extensions based on intersective levin classes. In *Annual Meeting of the Association for Computational Linguistics*.

Andrea Di Fabio, Simone Conia, and Roberto Navigli. 2019. VerbAtlas: a novel large-scale verbal semantic resource and its application to semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 627–637, Hong Kong, China. Association for Computational Linguistics.

Andrew Drozdov, Jiawei Zhou, Radu Florian, Andrew McCallum, Tahira Naseem, Yoon Kim, and Ramón Astudillo. 2022. Inducing and using alignments for transition-based AMR parsing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1086–1098, Seattle, United States. Association for Computational Linguistics.

Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime Carbonell. 2016. CMU at SemEval-2016 task 8: Graph-based AMR parsing with infinite ramp loss. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1202–1206, San Diego, California. 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.

Ana-Maria Giuglea and Alessandro Moschitti. 2006. Semantic role labeling via framenet, verbnet and propbank. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL - ACL '06*.

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.

James Gung. 2020. *Abstraction, Sense Distinctions and Syntax in Neural Semantic Role Labeling*. Ph.D. thesis, University of Colorado at Boulder.

James Gung and Martha Palmer. 2021. Predicate representations and polysemy in VerbNet semantic parsing. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 51–62, Groningen, The Netherlands (online). Association for Computational Linguistics.

Kuan-Hao Huang, Varun Iyer, Anoop Kumar, Sriram Venkatapathy, Kai-Wei Chang, and Aram Galstyan. 2022. Unsupervised syntactically controlled paraphrase generation with Abstract Meaning Representations. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1547–1554, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Ghazaleh Kazeminejad, Martha Palmer, Susan Windisch Brown, and James Pustejovsky. 2022. Componential analysis of english verbs. *Frontiers in Artificial Intelligence*, 5.

Alon Lavie. 2011. Evaluating the output of machine translation systems. In *Proceedings of Machine Translation Summit XIII: Tutorial Abstracts*, Xiamen, China.

Beth Levin. 1993. English verb classes and alternations: A preliminary investigation.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jungwoo Lim, Dongsuk Oh, Yoonna Jang, Kisu Yang, and Heuiseok Lim. 2020. I know what you asked: Graph path learning using AMR for commonsense reasoning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2459–2471, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yijia Liu, Wanxiang Che, Bo Zheng, Bing Qin, and Ting Liu. 2018. An AMR aligner tuned by transition-based parser. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2422–2430, Brussels, Belgium. Association for Computational Linguistics.

Abelardo Carlos Martínez Lorenzo, Marco Maru, and Roberto Navigli. 2022. Fully-semantic parsing and generation: the babelnet meaning representation. In *Annual Meeting of the Association for Computational Linguistics*.

Mathew Monfort, Bolei Zhou, Sarah Adel Bargal, Alex Andonian, Tom Yan, Kandan Ramakrishnan, Lisa M. Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, and Aude Oliva. 2018. Moments in time dataset: One million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:502–508.

Tahira Naseem, Austin Blodgett, Sadhana Kumaravel, Tim O'Gorman, Young-Suk Lee, Jeffrey Flanigan, Ramón Astudillo, Radu Florian, Salim Roukos, and Nathan Schneider. 2022. DocAMR: Multi-sentence AMR representation and evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3496–3505, Seattle, United States. Association for Computational Linguistics.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018a. 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.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018b. 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. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Terence Parsons. 1990. *Events in the semantics of English: A study in subatomic semantics*. MIT Press, Cambridge, MA.

James Pustejovsky. 1995. *The Generative Lexicon*. MIT Press, Cambridge, MA.

James Pustejovsky and Jessica Moszkowicz. 2011. The qualitative spatial dynamics of motion. *The Journal of Spatial Cognition and Computation*.

Kyeongmin Rim, Jingxuan Tu, Bingyang Ye, Marc Verhagen, Eben Holderness, and James Pustejovsky. 2023. The coreference under transformation labeling dataset: Entity tracking in procedural texts using event models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12448–12460, Toronto, Canada. Association for Computational Linguistics.

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

Karin Kipper Schuler, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *AAAI/IAAI*.

Karin Kipper Schuler, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of english verbs. *Language Resources and Evaluation*, 42:21–40.

Linfeng Song and Daniel Gildea. 2019. SemBleu: A robust metric for AMR parsing evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4547–4552, Florence, Italy. Association for Computational Linguistics.

Katharina Stein, Lucia Donatelli, and Alexander Koller. 2023. From sentence to action: Splitting amr graphs for recipe instructions.

Kevin Stowe, Jenette Preciado, Kathryn Conger, Susan Windisch Brown, Ghazaleh Kazeminejad, and Martha Palmer. 2021. Semlink 2.0: Chasing lexical resources. In *30th Annual Meeting of the Association for Computational Linguistics*, pages 222–227, Gronigen, Netherlands. Association for Computational Linguistics.

Haibo Sun and Nianwen Xue. 2024. Anchor and broadcast: An efficient concept alignment approach for evaluation of semantic graphs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*.

Christopher Tam, Richard Brutti, Kenneth Lai, and James Pustejovsky. 2023. Annotating situated actions in dialogue. In *Proceedings of the 4th International Workshop on Designing Meaning Representation*.

Jingxuan Tu, Eben Holderness, Marco Maru, Simone Conia, Kyeongmin Rim, Kelley Lynch, Richard Brutti, Roberto Navigli, and James Pustejovsky. 2022a. Semeval-2022 task 9: R2VQ – competence-based multimodal question answering. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Jingxuan Tu, Kyeongmin Rim, Eben Holderness, Bingyang Ye, and James Pustejovsky. 2023. Dense paraphrasing for textual enrichment. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 39–49, Nancy, France. Association for Computational Linguistics.

Jingxuan Tu, Kyeongmin Rim, and James Pustejovsky. 2022b. Competence-based question generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1521–1533, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jens E. L. Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, Jan Hajič, James H. Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, Rosa Vallejos, and Nianwen Xue. 2021. Designing a Uniform Meaning Representation for Natural Language Processing. *KI - Künstliche Intelligenz*, 35(3):343–360.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723.

Yuqing Yang, Qipeng Guo, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023. An AMR-based link prediction approach for document-level event argument extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12876–12889, Toronto, Canada. Association for Computational Linguistics.

Zixuan Zhang, Nikolaus Parulian, Heng Ji, Ahmed Elsayed, Skatje Myers, and Martha Palmer. 2021. Fine-grained information extraction from biomedical literature based on knowledge-enriched Abstract Meaning Representation. 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 6261–6270, Online. Association for Computational Linguistics.

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. Appendix

A.1. Prompts

We show the prompts for paraphrase generation of AMR and GLAMR graphs in Figure 4 and 5.

Task instruction:
Based on the information from previous sentence (not available if the current is the first sentence), the AMR graph of the current sentence as the context, generate an overall paraphrase in natural language focusing on the paraphrase of the implcit entities and entity state change from the subevents. Generate the paraphase as a separate paragraph.

Previous sentence:
[sentence text]

AMR graph:
[graph in penman notation]

Paraphrase:

Figure 4: GPT prompt for generating paraphrases from AMR graph.

Context knowledge:
GLAMR is an AMR-based graph with new roles and concepts that represent the implicit objects and subevent structure from the verbnet. For example, the new role ":imp_ouput" means the implicit output from the event, which needs to be paraphased into natural language. DocGLAMR is the supplementary graph to each sentence-level GMAR graph that provides anaphoric relations between entities from current and previous setences. Relations include aggregation (:aggregation), separation (:separation), transformation (:cut) and identity (:cui).

Task instruction:
Based on the information from previous sentence (not available if the current is the first sentence), the GLAMR and DocGLAMR graph of the current sentence as the context, generate an overall paraphrase in natural language focusing on the paraphrase of the implcit entities and entity state change from the subevents. Generate the paraphase as a separate paragraph.

Previous sentence:
[sentence text]

GLAMR graph:
[graph in penman notation]

DocGLAMR graph:
[graph in penman notation]

Figure 5: GPT prompt for generating paraphrases from GLAMR graph.