CxGr-AMR: Extending Abstract Meaning Representation Beyond Lexically Anchored Relations with Constructional Rolesets

Claire Bonial[1], **Claire Benet Post**[2], **Paul Van Eecke**[3], **Katrien Beuls**[4], **Harish Tayyar Madabushi**[5]
[1]DEVCOM U.S. Army Research Laboratory, [2]University of Colorado,
[3]Vrije Universiteit Brussel, [4]Université de Namur, [5]University of Bath
claire.n.bonial.civ@army.mil

Abstract

Current Abstract Meaning Representation (AMR) annotation guidelines, which largely tie argument structure to lexical rolesets, systematically misrepresent cases in which key semantic roles stem from clause-level structure rather than the verb, leaving these meanings either unnaturally attached, incorrect, or unexpressed. To address this limitation, we present CxGr-AMR, a novel extension of AMR that captures the semantics of various types of phrasal constructions, including argument structure constructions. We first examine how such cases are handled under current Standard-AMR guidelines and show that these analyses are often inadequate when constructionally contributed roles clash with those assigned by the verb. We then provide a theoretical grounding for our CxGr-AMR rolesets that lay out the relationship between the syntactic signatures of constructional slots and particular semantic roles associated with them. Finally, we develop an annotation-expert-in-the-loop pipeline for the semi-automatic annotation of sentences, and release a dataset containing 355 instances of phrasal constructions annotated with both Standard and CxGr-AMR.

Keywords: CxGr-AMR, Abstract Meaning Representation (AMR), constructional semantics, argument structure constructions

1. Introduction

When hearing or reading sentences such as "*Firefighters cut a three-year-old free*," or "*The coach shouted their players into a queue*," English speakers immediately grasp that there is more to the meaning of these sentences than the mere cutting or shouting events they evoke. One can indeed understand from the first sentence that it was a cutting action performed by firefighters that *resulted in* the freedom of a three-year-old, while the second sentence evokes a scenario where players *moved* into a queue as a result of the coach's shouting. The observation that there exists meaning that is contributed by larger syntactic patterns, which is not reducible to the meanings of the lexical items that instantiate them, forms a basic tenet of the theory of Construction Grammar (CxG) (Fillmore, 1988; Goldberg, 1995; Kay and Fillmore, 1999; Fried and Östman, 2004).

Abstract Meaning Representation (AMR) is a widely adopted, graph-based meaning representation with an established annotation system and broad downstream use (Banarescu et al., 2013), so failures of coverage have direct consequences for both annotation quality and semantic parsing. Although AMR has been used to represent the meaning of constructions in computational construction grammar implementations (e.g., Schmalz and Cornillie, 2022; Beuls and Van Eecke, 2025), AMR primarily anchors argument structure in lexical verbal "rolesets", and therefore struggles to represent the clause-level meaning in sentences such as those presented above without mis-attaching roles or leaving key meaning implicit.

This paper presents an extension of AMR that facilitates the annotation of semantic roles that are evoked by linguistic structures above the level of lexical items.[1] Crucially, AMR's inventory of senses and rolesets is designed to be extensible, so the missing clause-level relations can be added in a way that remains compatible with existing AMR graphs and tools. Extending AMR therefore lets us address this coverage gap without abandoning the standard formalism or its resources (see §3). Concretely, the extension we present enables accurate semantic parsing of phrasal constructions such as the Resultative: "*The man shrieked himself unconscious*." Examples (1) and (2) below illustrate the current Standard-AMR parse and the new CxGr-AMR representation for this sentence:

(1)
```
(s / shriek-01
      :ARG0 (m / man
                 :ARG1-of s)
      :mod (u / unconscious))
```

Parsed as: *The man shrieked, the thing-shrieked was the man, and the shrieking was unconscious*.

[1]We release the CxGr-AMR dataset, the constructional roleset specifications, and the annotation guidelines: https://github.com/H-TayyarMadabushi/cxgr-amr-construction-grammar

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 1–19
11 May 2026.

(2)
```
(r / resultative-91
    :ARG0 (m / man)
    :ARG1 m
    :ARG2 (u / unconscious)
    :ARG3 (s / shriek-01
        :ARG0 m))
```

Parsed as: *The man's shrieking caused him to become unconscious*.

Although many sentences can be accurately parsed according to verbal semantics alone, a comprehensive meaning representation should have the machinery to represent all meaning-bearing elements of a language. Furthermore, argument structure constructions constitute some of the most frequent constructions of English, and they are cross-linguistically common as well (Goldberg, 1995; Perek and Lemmens, 2010), yet any instantiation with a relatively infrequent, creative verb that does not share the semantics of the construction will not be represented accurately under current guidelines.

To address this gap, we present CxGr-AMR,[2] which extends the AMR formalism according to the theoretical tenets of CxG. After providing background information on AMR and CxG (§2), as well as our choice of the AMR formalism (§3), we detail the shortcomings of the current Standard-AMR representation of four English Argument Structure Constructions (ASCs) (§4): (1) Resultative, (2) Caused Motion, (3) Intransitive Motion, and (4) Ditransitive. In each subsection of §4, we then provide the proposed CxGr-AMR roleset. We describe how we develop a semi-automatic, expert-annotator-in-the-loop pipeline for the creation of a large corpus of CxGr-AMR (§5).

2. Background & Related Work

2.1. Standard-AMR

AMR represents the meaning of a sentence in the form of a graph (Banarescu et al., 2013). The "abstract" nature of AMR is intended to capture concepts and the relations between them devoid of idiosyncratic syntactic differences. Thus, for example, the realizations of "*fear*" as a verb or noun, and the adjective "*afraid*" are all annotated identically in AMR. This makes AMR an appealing formalism for representing meaning agnostic to the language and linguistic realization (Xue et al., 2014).

The edges of the graph represent semantic relations. AMR leverages the PropBank (Palmer et al., 2005) lexicon of "rolesets": for a given relation, the set of semantic roles licensed by that relation onto its syntactic arguments. The most frequent, and often "core" arguments of a relation are given argument numbers: ARG0-5 (Bonial et al., 2012).

AMR has an extensive set of relations that can be flexibly combined with any relation, such as the basic `:mod` or "modifier" role used in (1) to represent the meaning of "*unconscious*" with respect to the shrieking event.[3]

2.2. Lexical Bias in Standard-AMR

Although neither PropBank nor AMR explicitly embrace any theoretical viewpoint, the fact that most relations in the shared PropBank/AMR lexicon are individual lexical items leads to the implicit assumption that a single lexical head projects or licenses the semantic roles and argument structure of a clause. This approach is adequate for many sentences, wherein the semantics and argument structure of the verb are compatible with, and align with, the semantics and argument structure of the broader syntactic environment the verb is found in.

However, such an approach becomes problematic when facing cases wherein the lexical verb's senses are not commonly observed to license the arguments of the surrounding syntactic environment (first pointed out with respect to light verb annotation (Bonial and Palmer, 2016)). For example, the verb "*shout*," in its communication sense, commonly licenses a shouter and an utterance shouted, as well as an addressee shouted at or to. In the sentence "*The coach shouted their players into a queue*", the shouting event itself shares semantics with the communication sense, and yet the direct object is "*their players*" (not the utterance) and the prepositional phrase is the final goal location "*into a queue*" (not the addressee).

From a CxG theoretical approach, patterns of syntactic slots can assign semantic roles to the lexical items situated within them (Goldberg, 1995). Thus, the shouting example can be understood as an instantiation of the Caused Motion construction, which carries the motion semantics and assigns the thing-moved role as well as the goal location. If one instead posits that constructions can assign the semantic roles of a pattern of syntactic slots, then this vastly reduces the requisite size of the database of relations (Bonial et al., 2017). CxGr-AMR operationalises this idea within AMR by introducing rolesets that encode these clause-level roles *directly*, while preserving the verb's semantics as a concomitant event.

2.3. Data: The CoGS corpus

To evaluate the Standard-AMR treatment of constructions as well as motivate the requirements of

[2]Pronounced "Construction Grammar." The intention of this title is not to suggest that a grammar is encoded in AMR, but rather that we extend AMR in a manner following CxG approaches.

[3]AMR guidelines: https://github.com/amrisi/amr-guidelines

Construction	Meaning Description	Form Description	Example
Caused Motion	Agent of the action denoted by the verb causes theme to move along or towards a goal.	PHONOLOGY: /A_1 B_2 C_3 $D_{4/5}$/ MORPHOSYNTAX: [SBJ_1 [V_2 OBJ_3 OBL_4]$_{VP}$]$_5$	[[Workers]$_1$ dumped$_2$ [large burlap sacks of the imported material]$_3$ [into a huge bin. . .]$_4$]$_5$
Intransitive Motion	A theme carries out an event that causes or accompanies movement.	PHONOLOGY: /A_1 B_2 $C_{3/4}$/ MORPHOSYNTAX: [SBJ_1 [V_2 OBL_3]$_{VP}$]$_4$	[[The cyclone]$_1$ was sweeping$_2$ [across the state . . .]$_3$]$_4$
Ditransitive	Agent of the action denoted by the verb is construed as (intending to) cause a recipient to receive a theme.	PHONOLOGY: /A_1 B_2 C_3 $D_{4/5}$/ MORPHOSYNTAX: [SBJ_1 [V_2 OBJ_3 OBJ_4]$_{VP}$]$_5$	[[My ex]$_1$ feeds$_2$ [my kids]$_3$ [cheese whiz and R.C. Cola.]$_4$]$_5$
Resultative	Agent of the action denoted by the verb causes a patient to change / become a resulting state.	PHONOLOGY: /A_1 B_2 C_3 $D_{4/5}$/ MORPHOSYNTAX: [SBJ_1 [V_2 OBJ_3 OBL_4]$_{VP}$]$_5$	[[It]$_1$ jerks$_2$ you$_3$ awake$_4$ with the first sentence. . .]$_5$

Table 1: Excerpt of the argument structure construction descriptions that we embrace to develop our CxGr-AMR rolesets, drawn from Bonial and Tayyar Madabushi (2025).

the CxGr-AMR formalism, we leverage the Construction Grammar Schematicity (CoGS) corpus (Bonial and Tayyar Madabushi, 2025). The CoGS corpus is made up of about 600 instances of phrasal constructions, roughly equally distributed across 10 construction types (the Resultative construction is more frequent, given corpus expansion described in Scivetti et al. (2025)), including the four ASCs of focus in the present research. See Table 1 for CoGS ASC definitions and examples. The CoGS corpus is unique and particularly useful for this research because it is specifically made up of instances of constructions wherein the verbal semantics alone cannot account for the broader semantics.

This makes CoGS an ideal testbed for exploring a meaning representation's coverage of semantics. In other corpora, Standard-AMR may seem perfectly adequate because of the frequency with which ASCs are instantiated by a single, compatible lexical verb (such as "give" in the case of the Ditransitive, "put" in the case of Caused Motion, "make" in the case of the Resultative, and "go" in the case of Intransitive Motion). In CoGS, the same constructions are instantiated creatively with verbs that do not normally license the arguments and respective semantic roles surrounding it. CoGS also abstracts away from the separate problem of construction detection in raw text. Without a resource that makes constructionally contributed roles explicit (and models trained on it), we should not expect reliable identification of these instances at scale, which is part of the motivation for CxGr-AMR.

There are different schools of CxG, which differ slightly in details of how constructions are defined, enumerated, related to one another, and how they interact (see Hoffmann and Trousdale (2013) for an overview). We embrace the definition of constructions as pairings of form and meaning, where the form may be at the morphological, lexical, or phrasal level (Hoffmann, 2022). While the present research is largely compatible with all schools of CxG, we draw most heavily on Goldberg (1995). That work lays out the syntactic slots and associated semantic roles of ASCs (see Table 2). Our constructional rolesets are directly inspired by these associations between constructional slots and semantic roles.

Syntactic Slots	Subject	Verb	Object	Oblique
Lexical Items	*Firefighters*	*cut*	*the child*	*free*
ASC Roles	Cause	—	Patient	Result State
Verb Roles	Agent	—	—	—

Table 2: Resultative construction diagram demonstrating how both ASCs and verb semantic roles fuse in *Firefighters cut the child free*.

3. Rationale for an AMR Extension

Resources like Standard-AMR and PropBank demonstrate that annotating the mapping between words (and subgraph structures or syntactic trees) and particular semantic roles enables systems trained on such annotations to identify who is doing what to whom, when, where, and why (Palmer et al., 2022). Our analyses of Standard-AMR parses of CoGS argument structure constructions show that, when semantic roles are treated as stemming only from lexical relations, parsers trained on such data misparse cases where the verb's roleset does not align with the clause pattern.

Nonetheless, AMR is flexible enough to be extended to ASCs when we simply introduce rolesets for phrasal constructions.[4] These rolesets allow us to define the basic semantics of the construction, as well as the semantic role of each slot of the construction. Furthermore, although the present research is restricted to English, because AMR eschews annotating with respect to the part of speech of a given concept, it is not tied to an English-centric syntactic realization of a construction.

Finally, although the present research is the first exploration of Standard-AMR treatment of ASCs, there is a tradition of positing some rolesets for a limited set of English constructions in AMR. Past research has established rolesets for phrasal constructions including the Comparative Correlative (Bonial et al., 2018). However, past research assumes that verbal lexical relations are adequate for ASCs, whereas we provide evidence challenging this assumption.

4. ASCs: Problems & Solutions

In each section to follow, we analyze an individual ASC and show where the current Standard-AMR formalism cannot adequately capture the semantics of these constructions. We then outline the new CxGr-AMR roleset.

4.1. Resultative

The Resultative construction involves an Agent's participation in one event that causes a change of state in a Patient. As shown in Table 1, the Resultative construction is made up of 4 fully flexible (i.e. there are no fixed words or phonological forms) constructional slots:

PHONOLOGY: /A_1 B_2 C_3 $D_{4/5}$/
MORPHOSYNTAX: [SBJ_1 [V_2 OBJ_3 OBL_4]$_{VP}$]$_5$

The subject (A) is the agent of some event expressed by the verb (B), wherein carrying out that event causes the object (C) to change to the resulting state expressed by the oblique (D).

4.1.1. Standard-AMR Treatment: Resultative

Resultatives are particularly problematic within a Standard-AMR treatment first because AMR generally assesses arguments according to the projection of roles by a lexical verb relation, and second because Standard-AMR lacks a modifier role for results or secondary predication often expressed by adjectives in the Resultative. Thus, our assessment of Standard-AMR parses of CoGS Resultatives demonstrates a common pattern of misrepresenting the object of the lexical verb as a theme or patient of that verb and the expression of the resulting state as a modifier of the lexical verb. For example, the instance *The man shrieked himself unconscious* is parsed as seen in Example (1) (§1) according to the semantics of the shriek-01 roleset:[5]

ARG0-PAG: shrieker
ARG1-PPT: shriek itself, or utterance
ARG2-GOL: unfortunate listener

What the parse in (1) represents with respect to the meaning of this sentence is that the man is both the shrieker (ARG0) and the utterance (ARG1), and that the shrieking event is modified by *unconscious*. Thus, the parse incorrectly analyzes the false object as the patient and does not provide an informative analysis of the role of the resulting state.

While the automatically parsed data consistently exhibits this pattern of misrepresentation, which is rooted in Standard-AMR's anchoring of argument structure in lexical rolesets, we also consider whether Standard-AMR has in principle a way to represent Resultative semantics even when current parsers fail to recover it from existing corpora. At best, one can introduce a generic causal relation (e.g., cause-01) to relate the concomitant event (shriek-01) to the ensuing state (unconscious-01):

```
(c / cause-01
    :ARG0 (s / shriek-01
              :ARG0 (m / man))
    :ARG1 (u / unconscious-01
              :ARG1 m))
```

However, this representation collapses a distinction that English overtly encodes in form: Resultatives package the causing event and the resultant state as a single constructional causal complex, whereas "because" clauses encode a much broader class of causal/explanatory relations. Importantly, the contrast is not that "because" cannot be temporally overlapping (indeed it can), but that cause-01 alone does not mark (i) the event-internal/means-like construal characteristic of Resultatives (the shrieking is construed as the mechanism by which the change of state comes about), (ii) the constructional linking that tightly couples a participant in the concomitant event with the undergoer of the resultant state, and (iii) the result entailment/culmination that motivates the Resultative form. As a consequence, a graph using only cause-01 does not preserve why a speaker chose a Resultative rather than an explanatory "because" relation, and it provides no dedicated locus for construction-specific constraints that downstream parsing or generation would need to recover the intended construal.

4.1.2. CxGr-AMR: Resultative

To capture these constructional slots and their semantics, we develop the constructional roleset:

[4] See Appendix A for a discussion of other relevant NLP resources and AMR extensions.

[5] All rolesets are copied verbatim from https://propbank.github.io/v3.4.0/frames/.

Resultative-91

Arg0-Cause (SBJ slot (A))
Arg1-Patient (OBJ slot (C))
Arg2-Result (OBL slot (D))
Arg3-Concomitant_Event (V slot (B))

This roleset enables mapping the constructional slots (expressed as subgraphs) to the semantic roles of the construction. It leaves underspecified the precise nature of the verbal semantics within the construction, as this can vary. Additionally, we note that the Arg0 can vary in its agentivity depending upon the verb in the structure. For example, in *He knocked himself unconscious*, while *he* is the prototypical agent of the sentence (in the sense of Dowty (1991)), he is likely not intentional in this act. Example (2) (§1) shows the CxGr-AMR analysis of the example utterance *The man shrieked himself unconscious* discussed above.

4.2. Caused Motion

The Caused Motion construction involves an agent doing something that causes a theme to move along a path or towards some goal. The Caused Motion construction also involves four fully flexible or schematic slots:

PHONOLOGY: /A_1 B_2 C_3 $D_{4/5}$/
MORPHOSYNTAX: [SBJ_1 [V_2 OBJ_3 OBL_4]$_{VP}$]$_5$

The subject (A) is the agent of an event expressed by the verb (B), wherein carrying out that event causes the object (C) to move along some path or to a destination/goal expressed as an oblique (D).

4.2.1. Standard-AMR: Caused Motion

Standard-AMR has a fairly comprehensive inventory of roles that capture motion semantic roles including `:source`, `:path`, `:direction`, and `:destination`. Thus, if the lexical verb involved in the Caused Motion construction does involve any motion semantics, then the Standard-AMR treatment is potentially adequate. However, if the lexical verb of the Caused Motion construction does not typically involve any motion semantics, then Caused Motion semantics are often misanalyzed because the syntactic object of the lexical verb is construed as a prototypical argument of that verb in its typical sense. For example, the sentence *They laughed the actor off the stage* is automatically parsed according to the laugh-01 roleset:

ARG0-PAG: laugher
ARG1-PPT: cognate object
ARG2-CAU: source of joy
ARG3-PRD: end state of Arg0, as result of laughing

```
(z0 / laugh-01
    :ARG0 (z1 / they)
    :ARG2 (z2 / person
              :ARG0-of (z3 / act-01))
    :location (z4 / off
              :op1 (z5 / stage)))
```

The above parse denotes that they (ARG0) are laughing at the actor (ARG2), and that this laughter is happening off-stage (:location). Thus, this parse fails to capture any motion semantics at all.

One could capture Caused Motion instances with non-motion verbs using the introduction of causation and potentially introducing the implicit element of motion (i.e. they laughed, causing the actor to move off-stage):

```
(c / cause-01
  :ARG0 (l / laugh-01
         :ARG0 (t / they))
  :ARG1 (m / move-01
        :ARG1 (p / person
               :ARG0-of (a / act-01))
        :ARG2 (o / off
               :op1 (s / stage))))
```

Like the Resultative treatment, however, we note that this fails to capture the tightly coupled, complex-event nature of the causal and temporal relations between the motion and the concomitant event of laughter.

4.2.2. CxGr-AMR: Caused Motion

To capture the constructional slots and their semantics, we develop the following roleset:

Caused-Motion-91

Arg0-Cause (SBJ slot (A))
Arg1-Theme (OBJ slot (C))
Arg2-Goal (OBL slot (D))
Arg3-Concomitant_Event (V slot (B))

This roleset similarly supports mapping the constructional slots to the semantic roles of the construction, as exemplified by the newly annotated example *They laughed the actor off the stage*:

```
(c / caused-motion-91
    :ARG0 (t / they)
    :ARG1 (p / person
        :ARG0-of (a / act-01))
    :ARG2 (o / off
        :op1 (s / stage))
    :ARG3 (l / laugh-01
        :ARG0 t))
```

This represents the meaning: Their laughing caused the actor to move off the stage.

4.3. Intransitive Motion

The Intransitive motion construction involves a Theme in an event that causes or accompanies movement in space. The Intransitive Motion construction consists of three schematic constructional slots:

PHONOLOGY: /A_1 B_2 $C_{3/4}$/
MORPHOSYNTAX: [SBJ_1 [V_2 OBL_3]$_{VP}$]$_4$

The subject (A) is a theme carrying out an event (B) that causes or accompanies movement along a path expressed as an oblique (C).

4.3.1. Standard-AMR: Intransitive Motion

As described for the Caused Motion construction, the inventory of motion-related modifier roles within the Standard-AMR inventory enables reasonable annotation of Intransitive Motion construction in cases where the lexical verb can felicitously be construed as compatible with such motion arguments. In these cases, the lexical relation roleset also reflects motion semantics, so that the argument structure of the lexical verb aligns for one or more arguments with the expected arguments of the Intransitive Motion construction.

In cases where the lexical verb is not semantically compatible with such motion modifiers, such as sound emission verbs, the Standard-AMR parser output misrepresents the Intransitive Motion meaning. For example, *...troops rumbled along the main road...* is annotated with the following roleset for the sound emission verb rumble-01 and assigned the following parse:

ARG0-PPT: entity rumbling
ARG1-PPT: sound/utterance
ARG2-GOL: hearer

```
(z0 / rumble-01
    :ARG0 (z5 / troop
    :location (z8 / along
            :op1 (z9 / road
                    :mod (z11 / main))))
```

This parse fails to capture that the troops are in motion at all. However, this sentence is assigned a gold standard parse in the AMR corpus and is exemplified in Bonial et al. (2018), where the `:location` above is simply swapped for a `:path` modifier. Thus, the Standard-AMR machinery can posit the path argument, but fails to account for the fact that the sound emission lexical verb would not generally license motion arguments.

4.3.2. CxGr-AMR: Intransitive Motion

We represent the constructional slots and semantics of the Intransitive Motion construction with the following roleset:

Intransitive-Motion-91
Arg1-Theme (SBJ slot (A))
Arg2-Path/Goal (OBL slot (C))
Arg3-Concomitant_Event (V slot (B))

This roleset can be applied to the instance *Troops rumbled along the main road* to give the following CxGr-AMR graph:

```
(i / intransitive-motion-91
    :ARG1 (t / troop
    :ARG2 (a / along
            :op1 (r / road
                    :mod (m / main)))
    :ARG3 (r2 / rumble-01
            :ARG0 t))
```

This precisely captures the motion semantics of the construction: the troops move, rumbling, along the main road. Notice that the Intransitive Motion construction is a more parsimonious way of expressing this precise framing of the event, as opposed to the preceding rephrasing with "move."

4.4. Ditransitive

The Ditransitive construction involves an agent carrying out an event that is construed as causing a recipient to receive a theme. The construction consists of four fully schematic slots:

PHONOLOGY: /A_1 B_2 C_3 $D_{4/5}$/
MORPHOSYNTAX: [SBJ_1 [V_2 OBJ_3 OBJ_4]$_{VP}$]$_5$

The subject (A) carries out an action denoted by the verb (B), which is construed as (intending to) cause a recipient, expressed as the object (C), to receive a theme, expressed as a second object (D).

4.4.1. Standard-AMR: Ditransitive

When analyzing Standard-AMR parses of CoGS Ditransitives, we see again the pattern emerge wherein if the lexical verb within the Ditransitive is semantically compatible with the meaning of the Ditransitive, and therefore one or more arguments of the lexical verb align with expected arguments of the Ditransitive, then the Standard-AMR parse is generally adequate, although perhaps lacking the precise semantics. Standard-AMR also includes a `:beneficiary` that can be used in cases where the lexical verb roleset does not include a recipient or beneficiary argument.

However, we again see the pattern that verbs that are not typically found in a Ditransitive structure and would not generally license a recipient are given problematic parses in Standard-AMR. For example, *They're going to kill Reagan a commie* is assigned the kill-01 roleset and parsed in the following way:

ARG0-PAG: killer
ARG1-PPT: corpse
ARG2-MNR: instrument

```
(z0 / kill-01
    :ARG0 (z1 / they)
    :ARG1 (z2 / person
            :wiki "Ronald_Reagan"
            :name (z3 / name
```

```
                     :op1 "Reagan")
          :mod (z4 / commie)))
```

This parse denotes that they (ARG0) kill Reagan (ARG1), who is modified as communist. This demonstrates how automatic parsers trained without regard to constructional semantics can wholly misrepresent the semantics of a sentence.

4.4.2. CxGr-AMR: Ditransitive

To represent the meaning of the Ditransitive, we develop the following roleset:

Ditransitive-91

Arg0-Cause (SBJ slot (A))
Arg1-Theme (OBL slot (D))
Arg2-Recipient (OBJ slot (C))
Arg3-Concomitant_Event (V slot (B))

When applied to our earlier, problematic example, we now derive the following parse:[6]

```
(d / ditransitive-91
   :ARG0 (t / they)
   :ARG1 (c / commie)
   :ARG2 (p / person
            :wiki "Ronald_Reagan"
            :name (n / name
                     :op1 "Reagan"))
   :ARG3 (k / kill-01
            :ARG0 t
            :ARG1 c))
```

This correctly captures the relations among arguments as well as the (intended) Ditransitive transfer/beneficiary semantics.

4.5. Constructional Roleset Summary

The assignment of numbered arguments is closely paralleled for each constructional roleset. This is intentional, and the argument number assignment itself is guided by the PropBank guideline for Arg0 to be a prototypical agent, Arg1 to be a prototypical patient, and Arg2 to be another core argument, but what this argument is depends upon the class of the relation. For example, relations that lexically entail motion would be a goal, while those that entail transfer have a recipient Arg2. Consistent numbering facilitates combining data for different argument types across relations, especially given that data might be sparse for any single relation. Thus, CxGr-AMR provides another source of data enabling understanding of how agents, patients, and motion-related arguments are realized in English because Arg0s, Arg1s and Arg2s across motion relations, now including phrasal constructions, can be combined.

[6]To reduce the size of the graph for this paper, we have simplified the structure of “commie”, which calls for representation using the `:have-org-role-91` roleset.

To maintain symmetry and consistency with the argument structures of existing relations, we shift the concomitant event slot to Arg3. Thus, within CxGr-AMR, consistency in role numbers allows one to generalize over, for example, concomitant events for a particular set of constructions. As we expand to additional constructions, this also facilitates potentially combining or splitting constructions. For example, it is possible to annotate Intransitive Resultatives with the Resultative-91 roleset (while simply omitting the Arg0), but alternatively, the intransitive cases could easily be detected automatically given their unique argument structure signature and assigned to a finer-grained roleset if desired.

Finally, we emphasize that the CxGr-AMR rolesets are not intended to encode English syntax or to elevate a particular surface pattern to the level of meaning representation. The syntactic “signatures” we reference throughout §4 are used only as a practical cue for where English reliably packages certain meanings; what the rolesets make explicit is the constructionally contributed semantics (e.g., result entailment, caused motion, caused transfer) that would otherwise be implicit or mis-attached under a purely lexically anchored analysis. In this sense, the constructional node functions as a typed semantic operator that remains compatible with Standard-AMR (lexical predicates are retained as concomitant events) while providing a uniform locus for clause-level meaning that can generalize beyond any single verb and, in principle, beyond English-specific realizations.

5. CxGr-AMR Corpus Development

Writing AMR graphs has been established as a notoriously time-consuming and expensive task (Martin et al., 2020). Banarescu et al. (2013) have asserted that, using the AMR annotator, annotation time averages around 7-10 minutes per sentence.[7] For our targeted CoGS corpus of more than 350 sentences, a large portion of which are lengthy (i.e. 20+ words), and would require additional annotation time, we sought to shorten the annotation process so that we could focus on the new constructions proposed in this paper.

To this end, we formulated a semi-automated annotation process composed of four parts: 1) Standard-AMR parsing for the initial graphs, 2) LLM-based editing to incorporate the updated rolesets, 3) rule-based logical fixes, and 4) manual post-editing of the resulting graphs. Using this system allowed us to bypass the expensive, start-from-scratch annotation step and move directly to the faster process of editing the generated graphs. We

[7]This estimate reflects the time for *trained* annotators. Annotators not extensively trained easily require 30 minutes to an hour.

note that while CoGS provides construction labels for our initial study, the longer-term goal is for CxGr-AMR to supply the supervision needed to learn these constructional meanings from raw text rather than assuming gold construction identification.

5.1. Automated Parsing

For the automated AMR parsing portion of this project, we utilized the SPRING parser (Bevilacqua et al., 2021), a state-of-the-art AMR parser with a SMATCH score (Cai and Knight, 2013) of 83.0 (Blloshmi et al., 2021) on the LDC2020T0 dataset (Knight et al., 2020).[8]

Next, we manually edited from the AMR parser graphs a small subsection of the data that included 12 sentences divided equally by construction type: Resultative, Caused Motion, Intransitive Motion, and Ditransitive. Each of the three sentences chosen included one simple sentence (wherein the head node that would become -91 roleset was not embedded within another clause), a complex sentence (wherein the -91 roleset was embedded within another clause), and a passive variant. This was done so that we would have a few in-context learning examples for each sentence type, thereby demonstrating how the -91 roleset should be translated from the Standard-AMR.[9]

Once these graphs were produced, we checked the graphs for common errors, including if the model mistakenly added two CxGr-AMR rolesets to the graph, if any of the variable names were duplicated, if a graph was not produced, or if there were not the same number of open parentheses to match closed parentheses in the graph.[10] The last error type (of parentheses issues) was the most common; a simple script fix was used to delete or add parentheses.

On the remaining set of graphs that still had errors after rule-based corrections, we ran these using the more expensive GPT 5.2 model. Most of the remaining problematic graphs were longer on average than those without obvious errors,[11] and we saw an improvement in graph quality with the larger model. Once these were run, the rule-based corrections script was run once more and parentheses were checked and fixed again.

[8]For a general overview of AMR parsers, please see the NLP progress leader-board that is available here: https://nlpprogress.com/english/semantic_parsing.html#amr-parsing.

[9]The exact prompts with these examples are in subsection 8.1 in Appendix B. The first run of corrections included these instructions and were run on GPT-5-mini.

[10]See Figure 1 in Appendix B for details on the number of these errors in the AMR GPT-5-mini graphs.

[11]See Figure 2 in Appendix B.

5.2. Manual Edits

After automatic processing, the graphs were manually reviewed. We divided the data by construction type so each of our four annotators could specialize in a single construction. Graphs for manual correction were presented in a spreadsheet that included a binary correctness judgment, a column for corrected graphs, and a notes column for cases requiring adjudication. Annotators were instructed to flag uncertain cases for group discussion, which we resolved through adjudication sessions. Annotators followed roleset specifications when adjusting concomitant events and avoided introducing roles not licensed by the frame index.[12]

Manual review revealed that while many graphs were structurally well-formed, semantic and roleset-level issues remained common. In one annotator's subset of Ditransitives, for example, 33/50 graphs contained at least one error. Frequent error types included incorrect argument layouts within rolesets (such as ARG2 instead of ARG3), selection of an incorrect roleset sense (such as *run-01* instead of *run-02*), hallucinated roles and rolesets, misplaced scope (such as temporal modifiers attached at the wrong level), and occasional spurious modifiers or wiki links. Pronoun normalization errors, like *I* vs. *me*, also appeared.

5.3. Corpus Statistics

In Table 3, we have summarized the current corpus. We note the correlation between average sentence length to graphs that required manual fixes after the automated annotation process. There is a clear trend that the auto-generated graphs suffer in quality given longer sentences, thereby requiring more manual fixes.[13] Additionally, given the high percentages of graphs that were fixed, manual checking and correction is not a step that can be bypassed: when 69.2% of Resultative graphs were not generated correctly, it is clear that a human still needs to be in the annotation loop.

6. Future Work

Our annotation experiments indicate that fully automatic conversion remains unreliable. This motivates future work aimed at improving annotation support and automatic processing for these constructions. Therefore, we plan to develop dedicated annotation tools for CxGr-AMR and to extend it to other constructions beginning with those in the CoGS corpus (see Figure 4 in Appendix B). As we develop these tools and conduct manual

[12]Rolesets found here: https://propbank.github.io/v3.4.0/frames/.

[13]A visual is available in Figure 3 in Appendix B.

CxN Type	Count	% Fix	LenGPT	LenFix	LenAll
Resultative	192	69.2	8.92	12.56	17.27
Intransitive	58	35.3	12.09	19.17	22.05
Caused Motion	55	50	16.14	21.86	24.11
Ditransitive	50	66	7.29	19.67	15.46
Total	355	60.8	10.15	17.21	18.86

Table 3: Construction type, count, percentage of graphs needing a manual fix, avg sentence length of graphs that did *not* need a manual fix, avg sentence length of graphs needing manual fix, and overall avg sentence length.

edits more broadly, we will also conduct broader inter-annotator-agreement measures of CxGr-AMR. We will also evaluate whether CxGr-AMR improves generalization in downstream tasks that depend on explicit argument structure.

In the future, we will explore how these methods can support cross-linguistic meaning representation, particularly for languages where morphosyntax and clause patterns contribute important semantic roles that are not well captured by lexical rolesets alone (see Appendix C for an initial exploration within Quechua).

One natural extension is to Uniform Meaning Representation (UMR) (Van Gysel et al., 2021), which builds on AMR's PropBank backbone while adding aspect, modality, discourse structure, and a more abstract inventory of participant roles in order to be more cross-linguistically applicable. While UMR introduces generalized roles[14] such as *actor*, *undergoer*, and *theme*, these remain largely tied to predicate-level argument structure and do not fully address meaning contributed by constructions. Cross-linguistically, many semantic roles arise from morphosyntactic patterns (e.g., caused motion, applicatives, valency-changing morphology) rather than from the lexical predicate itself.

Our approach complements UMR by introducing constructional rolesets that explicitly encode these patterns independently of any single predicate. Importantly, this also yields a more efficient and interpretable representation, where constructional meaning is made explicit rather than implicitly distributed across predicate-specific roles. This is particularly useful in languages where PropBank-style rolesets are incomplete or unavailable, where the UMR roleset assignment system would be utilized, since these defined structures may be used if the concept exists in another language (see Appendix C for examples).

Our solution extends easily to English UMR, and the example below illustrates this integration in English while Appendix C provides further examples for Quechua: while the standard UMR graph encodes the event via *drive-01*, the constructional extension introduces an *intransitive-motion-91* node that captures motion semantics contributed by the construction, making the decomposition of meaning more transparent by separating path from manner.

(1) Standard AMR: *He drove through the tunnel.*

```
(d / drive-01
    :ARG0 (h / he)
    :path (t / tunnel))
```

(2) UMR (Bonn et al., 2024):

```
(d/ drive-01
    :ARG0 (p/ person
        :ref-person 3rd
        :ref-number Singular)
    :path (t/ tunnel)
    :aspect Performance)
```

(3) UMR + CxGr-AMR:

```
(m / intransitive-motion-91
    :ARG1 (p / person
        :ref-person 3rd
        :ref-number Singular)
    :ARG2 (r / through
        :op1 (t / tunnel))
    :ARG3 (d / drive-01
        :ARG0 p
        :aspect Performance))
```

7. Conclusions

AMR remains important even in the current landscape of LLMs, particularly in settings where meaning representations must be explicit and auditable. Yet Standard-AMR largely links core semantic roles to lexical rolesets, which leads to workarounds for clauses whose argument structure is not naturally licensed by the verb alone. This coverage gap matters because it affects very common and productive patterns of English, and it therefore limits the reliability of AMR-based semantic parsing.

We addressed this limitation by introducing CxGr-AMR, an extension of AMR that adds explicit rolesets for clause patterns that contribute semantic roles *beyond* those projected by the lexical predicate. We grounded these rolesets in existing linguistic analyses, and we released a dataset of 355 instances annotated in both Standard-AMR and CxGr-AMR, together with the roleset specifications and annotation guidelines. Compared to Standard-AMR workarounds that rely on generic causation or motion paraphrases, CxGr-AMR provides a more direct and transparent representation of meaning.

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

Limitations

Although the framework is designed with cross-linguistic applicability in mind (see Appendix C), the present study evaluates the approach on English data. Extending the framework to additional languages will require further investigation beyond the scope of this initial paper on how constructional semantics interact with language-specific patterns.

Another limitation concerns the annotation process. While part of the annotation pipeline was automated, a substantial portion (about 60%) of graphs required at least one manual correction; thus, it is clear that human annotators must remain in the loop to accurately capture these constructions, particularly when introducing new rolesets and structural conventions. As well, the semi-automatic pipeline relies on large language models for graph editing. While this approach substantially reduces valuable annotation time, it could introduce potential concerns related to reproducibility, model drift, and hallucinated structures. This fact should be taken into account by researchers interested in utilizing, reproducing, or extending this work.

8. Bibliographical References

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on 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.

Katrien Beuls and Paul Van Eecke. 2025. Construction grammar and artificial intelligence. In Mirjam Fried and Kiki Nikiforidou, editors, *The Cambridge Handbook of Construction Grammar*, pages 543–571. Cambridge University Press, Cambridge, United Kingdom.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 12564–12573.

Rexhina Blloshmi, Michele Bevilacqua, Edoardo Fabiano, Valentina Caruso, and Roberto Navigli. 2021. Spring goes online: end-to-end amr parsing and generation. In *Proceedings of the 2021 conference on empirical methods in natural language processing: system demonstrations*, pages 134–142.

Hans C Boas. 2021. Construction grammar and frame semantics. In *The Routledge handbook of cognitive linguistics*, pages 43–77. Routledge.

Claire Bonial, Bianca Badarau, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Tim O'Gorman, Martha Palmer, and Nathan Schneider. 2018. Abstract meaning representation of constructions: The more we include, the better the representation. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Claire Bonial, Kathryn Conger, Jena D Hwang, Aous Mansouri, Yahya Aseri, Julia Bonn, Timothy O'Gorman, and Martha Palmer. 2017. Current directions in english and arabic propbank. In *Handbook of linguistic annotation*, pages 737–769. Springer.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie 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.

Claire Bonial, Jena Hwang, Julia Bonn, Kathryn Conger, Olga Babko-Malaya, and Martha Palmer. 2012. English propbank annotation guidelines. *Center for Computational Language and Education Research Institute of Cognitive Science University of Colorado at Boulder*, 48:14.

Claire Bonial and Martha Palmer. 2016. Comprehensive and consistent propbank light verb annotation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3980–3985.

Claire Bonial and Harish Tayyar Madabushi. 2025. Constructing understanding: on the constructional information encoded in large language models. *Language Resources and Evaluation*, 59(4):4559–4598.

Julia Bonn, Matthew J Buchholz, Jayeol Chun, Andrew Cowell, William Croft, Lukas Denk, Sijia Ge, Jan Hajic, Kenneth Lai, James H Martin, Skatje Myers, Alexis Palmer, Martha Palmer, Claire Benét Post, James Pustejovsky, Kristine Stenzel, Haibo Sun, Zdeňka Urešová, Rosa Vallejos, Jens E L Van Gysel, Meagan Vigus, Nianwen Xue, and Jin Zhao. 2024. Building a broad infrastructure for uniform meaning representations. In *Proceedings of the 2024 Joint*

International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 2537–2547.

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.

David Dowty. 1991. Thematic proto-roles and argument selection. *language*, 67(3):547–619.

Charles J Fillmore. 1967. The case for case.

Charles J. Fillmore. 1988. The mechanisms of “construction grammar”. In *Annual Meeting of the Berkeley Linguistics Society*, volume 14, pages 35–55.

Mirjam Fried and Jan-Ola Östman. 2004. Construction grammar: A thumbnail sketch. In Jan-Ola Östman and Mirjam Fried, editors, *Construction grammar in a cross-language perspective*, pages 1–86. John Benjamins, Amsterdam, Netherlands.

Adele E. Goldberg. 1995. *Constructions: A construction grammar approach to argument structure*. University of Chicago Press, Chicago, IL, USA.

Thomas Hoffmann. 2022. *Construction grammar*. Cambridge University Press.

Thomas Hoffmann and Graeme Trousdale. 2013. *The Oxford handbook of construction grammar*. Oxford University Press.

Paul Kay and Charles Fillmore. 1999. Grammatical constructions and linguistic generalizations: The *What’s X Doing Y?* construction. *Language*, 75(1):1–33.

Kevin Knight, Bianca Badarau, and Laura Banarescu. 2020. *Abstract meaning representation (amr) annotation release 3.0*. Lead Discovery Center LDC.

Mary Martin, Cecilia Mauceri, Martha Palmer, and Christoffer Heckman. 2020. Leveraging non-specialists for accurate and time efficient amr annotation. In *Proceedings of the LREC 2020 Workshop on" Citizen Linguistics in Language Resource Development"*, pages 35–39.

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

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2022. Machine learning for semantic role labeling. In *Semantic Role Labeling*, pages 31–52. Springer.

Florent Perek and Maarten Lemmens. 2010. Getting at the meaning of the english at-construction: the case of a constructional split. *CogniTextes. Revue de l’Association française de linguistique cognitive*, 5(Volume 5).

Veronica Juliana Schmalz and Frederik Cornillie. 2022. Towards truly intelligent and personalized icall systems using fluid construction grammar. In *Colpaert, J., Wang, Y., & Stockwell, G.(Eds.)(2022). Proceedings of the XXIst International CALL Research Conference. London: Castledown Publishers. https://doi.org/10.29140/9781914291050*, pages 169–179. Castledown Publishers.

Wesley Scivetti, Melissa Torgbi, Mollie Shichman, Taylor Pellegrin, Austin Blodgett, Claire Bonial, and Harish Tayyar Madabushi. 2025. Beyond memorization: Assessing semantic generalization in large language models using phrasal constructions. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 1184–1201.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O’Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3):343–360.

Nianwen Xue, Ondrej Bojar, Jan Hajic, Martha Palmer, Zdenka Uresova, and Xiuhong Zhang. 2014. Not an interlingua, but close: Comparison of english amrs to chinese and czech. In *LREC*, volume 14, pages 1765–1772. Reykjavik, Iceland.

Appendix A: Related NLP Resources

There has been a close relationship between construction grammar and frame semantics as both are rooted in Fillmore’s case grammar (Fillmore, 1967) (see also Boas (2021).) A basic principle of CxG is that constructions carry meaning, not only lexical items: something that is typically not reflected in semantic formalisms. Specifically within NLP resources, corpora that are exhaustively annotated with semantics, such as PropBank, tie roleset instances to lexical units. Even FrameNet, a repository of frames based directly on Fillmore’s work, is

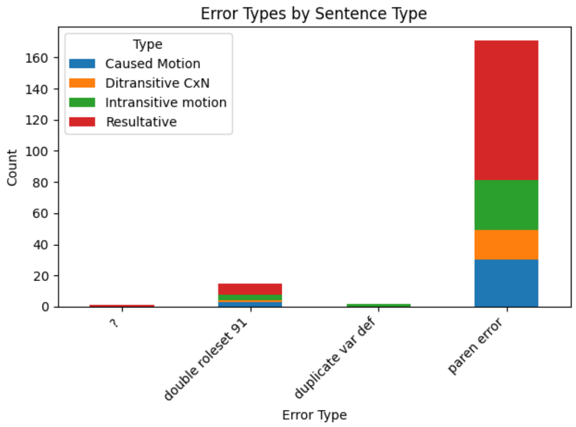

Figure 1: Initial errors found when looking at the GPT-5-mini auto-generated graphs that were made from the input of a sentence, a construction specific prompt, and the AMR SPRING parser graph.

modeled in such a way where frames are evoked by lexical units (Baker et al., 1998).

As we show throughout Section 4, Standard-AMR solutions are rather complex and lack generality. Thus, we vastly extend the initial efforts of encoding select constructional semantics into AMR Bonial et al. (2018) by leveraging dedicated rolesets on a more abstract level. CxGr-AMR is compatible with other AMR extensions, as each makes use of the Standard-AMR inventory of relations, but then adds other elements, such as speech acts in the case of Dialogue-AMR (Bonial et al., 2020), and cross-lingual meaning-bearing elements in the case of Uniform Meaning Representation (Van Gysel et al., 2021).

Appendix B: LLM Experiment Details

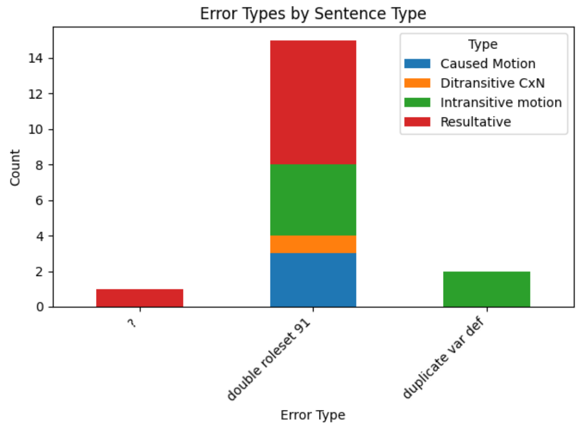

Figure 2: The errors that remained in the auto-generated graphs once the parentheses issues were removed with a simple script.

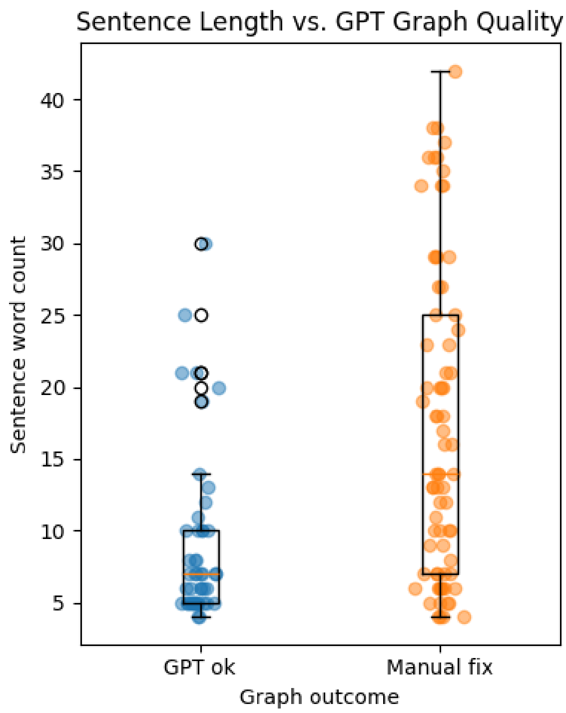

Figure 3: A scatter plot overlaid with a bar plot showing sentence counts for graphs generated by GPT that annotators decided did not need fixes, shown in blue, versus those that did need manual fixes, shown in orange.

8.1. GPT Prompt Examples

Below is each of the verbatim prompts used when prompting GPT models to edit SPRING parser graphs along with their sentences. The model was given a different prompt based on which construction type the sentence was classified under.

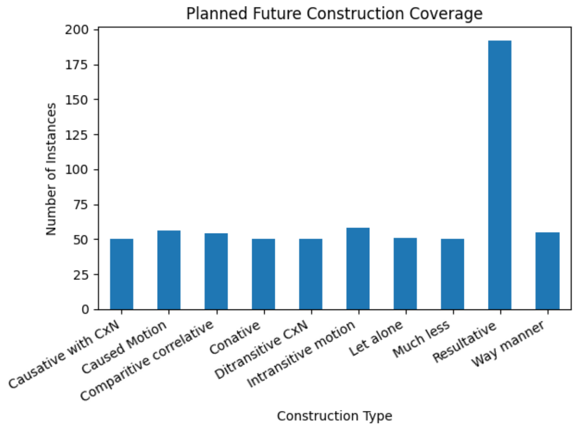

Figure 4: Counts of different construction types available in our corpus. Eventually, we would like to expand the dataset to all of these constructions.

8.1.1. Resultative

Update the AMR to use the constructional roleset resultative-91 when the sentence expresses a RESULTATIVE construction (an event causes an entity to end up in a resulting state/role/location).

I will give you a sentence and AMR graph with a resultative construction within it.

Example sentences Resultative Construction:
Could he shriek himself unconscious?
Firefighters cut the man free
He had often drunk himself silly

With various OBL types:
He wiped the table very clean. (ADJ)
Pop music drives him round the bend. (PP)
They elected him president. (NP)
A judge ordered the recordings to be made public. (nonfinite clause)

In the auto-generated AMR for the sentence "the man shrieked himself unconcious" we get:

```
(z0 / shriek-01
    :ARG0 (z1 / man
              :ARG1-of z0)
    :mod (z2 / unconscious))
```

This is a Resultative Construction wherein:
Sentence: The man shrieked himself unconscious
Shriek.01 has the arguments:
ARG0-PAG: shrieker
ARG1-PPT: shriek itself, or utterance
ARG2-GOL: unfortunate listener

This can be read as: "man is treated as shrieker and thing-shrieked, shrieking event is modified as unconscious". Which is problematic and not the actual MEANING of the sentence.

We want to change this graph to instead look like this:

```
(r / resultative-91
    :ARG0 (m / man)
    :ARG1 m
    :ARG2 (u / unconscious)
    :ARG3 (s / shriek-01
        :ARG0 m))
```

Wherein we have the Resultative.91 construction:

Resultative.91
Arg0-Cause: The man
Arg1-Patient: The man
Arg2-Result: unconscious
Arg3-Concomitant_Event: shriek.01
(Arg0: The man)

Here are some additional sentences with the orginal auto-generated AMR and the graph we transform them into:

Sentence: she laughed herself silly.

auto-amr:

```
(z0 / laugh-01
    :ARG0 (z1 / she)
    :manner (z2 / silly))
```

manually corrected AMR:

```
(r / resultative-91
    :ARG0 (s / she)
    :ARG1 s
    :ARG2 (s2 / silly)
    :ARG3 (l / laugh-01
        :ARG0 s))
```

Sentence: This nice man probably just wanted Mother to KISS him unconscious.

auto AMR:

```
(z0 / probable
    :domain (z1 / want-01
        :ARG0 (z2 / man
            :ARG1-of (z3 / nice-01)
            :mod (z4 / this))
        :ARG1 (z5 / kiss-01
            :ARG0 (z6 / person
                :ARG0-of (z7 /
                have-rel-role-91
                    :ARG1 z2
                    :ARG2
```

```
                    (z8 / mother)))
            :ARG1 z2
            :manner (z9 / unconscious))
        :mod (z10 / just)))
```

manually corrected AMR:

```
(p / probable-01
    :ARG1 (w / want-01
        :ARG0 (m / man
            :mod (t / this)
            :ARG1-of (n / nice-01))
        :ARG1 (r / resultative-91
            :ARG0 (p / person
                :name (n2 / name
                :op1 "Mother"))
            :ARG1 m
            :ARG2 (u /
            unconscious-02
                ARG1 m)
            :ARG3 (k / kiss-01
                :ARG0 p
                :ARG1 m))
        :mod (j / just)))
```

Sentence: I was scared stupid by what I saw.

auto AMR:

```
(z0 / scare-01
    :ARG0 (z1 / thing
        :ARG1-of (z2 / see-01
            :ARG0 (z3 / i
                :mod (z4 /
                stupid))))
    :ARG1 z3)"
```

manually corrected AMR:

```
(r / resultative-91
    :ARG0 (t / thing
        :ARG1-of (s / see-01
            :ARG0 (i / i)
            :ARG1 t))
    :ARG1 i
    :ARG2 (s2 / stupid)
    :ARG3 (s3 / scare-01
        :ARG0 t
        :ARG1 i))
```

8.1.2. Caused-motion

"I want you to update the graph to include the new roleset caused-motion-91.

Caused motion constructions occur when the Agent is doing something that causes theme to move along or towards a goal.

An example sentence with Caused Motion Construction:
They laughed the actor off the stage

Here the caused motion construction should be:

```
Caused-Motion.91:
 Arg0-cause: They
 Arg1-theme: the actor
 Arg2-goal: off the stage
 Arg3-concomitant-event: laugh.01
   (Arg0: They)
```

An example of where the auto-generated AMR fails to capture this structure is in the following instance:

The Caused Motion Construction is:
They laughed the actor off the stage

the auto generated graph is:

```
(z0 / laugh-01
    :ARG0 (z1 / they)
    :ARG2 (z2 / person
              :ARG0-of (z3 / act-01))
    :location (z4 / off
                  :op1 (z5 / stage)))
```

This graph has the head node as Laugh.01
(mismatch with caused-motion Arg structure)

```
 ARG0-PAG: laugher
 ARG1-PPT: cognate object
 ARG2-CAU: source of joy
 ARG3-PRD: end state of arg0, as result
 of laughing
```

Which semantically represents something like: "They laughed at the actor off-stage"

If we were to fix this to represent the actual meaning we would use caused-motion-91:

```
Caused-Motion.91
  Arg0-cause: They
  Arg1-theme: the actor
  Arg2-goal: off the stage
  Arg3-concomitant-event: laugh.01
    (Arg0: They)
```

Another Common Current AMR Treatment (Not-As-Bad): Caused-Motion
Sentence: After the seven hundred passengers on the train were stranded for two hours , they were GUIDED through the tunnels to a safe place .

auto AMR graph:

```
(z0 / guide-01
    :ARG1 (z1 / passenger
        :quant 700)
    :ARG2 (z2 / place
        :ARG1-of (z3 / safe-01))
    :path (z4 / tunnel)
    :time (z5 / after
        :op1 (z6 / strand-01
            :ARG1 z1
            :location (z7 / train)
            :duration (z8 /
            temporal-quantity
                :quant 2
```

```
            :unit (z9 /
            hour)))))
```

Here Guide.01 (aligns with caused-motion Arg structure):
ARG0-PAG: guide, agent
ARG1-PPT: entity guided
ARG2-GOL: guided in/through
ARG3-PRD: signposts along the way

We could fix it with Caused-Motion.91:
Arg0-cause: -
Arg1-theme: the seven hundred passengers
Arg2-goal: to a safe place
Arg3-concomitant-event: guide.01 (Arg0: They)

Here are some additional sentences with the orginal auto-generated AMR and the graph we want to transform them into:

Sentence: She wiggled her feet out of the boots.

auto-amr:

```
""(z0 / wiggle-01
    :ARG0 (z1 / she)
    :ARG1 (z2 / foot
        :part-of z1)
    :direction (z3 / out-of
        :op1 (z4 / boot)))""
```

manually corrected AMR:

```
""(c / caused-motion-91
    :ARG0 (s / she)
    :ARG1 (f / foot
             :part-of s)
    :ARG2 (g / out-of
             :op1 (b / boot))
    :ARG3 (w / wiggle-01
             :ARG0 s
             :ARG1 f))""
```

Sentence: The stone was THROWN across the river.

auto-amr:

```
""(z0 / throw-01
    :ARG1 (z1 / stone)
    :path (z2 / across
             :op1 (z3 / river)))""
```

manually corrected AMR:

```
""(c / caused-motion-91
    :ARG1 (s / stone)
    :ARG2 (a / across
        :op1 (r / river))
    :ARG3 (t / throw-01
        :ARG1 s))""
```

Sentence: Fundamentally, everyone is entitled to a private life and - no matter who they are - they have a right not to have their personal life DRAGGED through the mud for political point scoring or for the general consumption of the public.

auto-amr:

```
(z0 / and
    :op1 (z1 / entitle-01
        :ARG1 (z2 / life
            :ARG1-of (z3 /
            private-02))
        :ARG2 (z4 / everyone))
    :op2 (z5 / right-05
        :ARG1 z4
        :ARG2 (z6 / drag-01
            :polarity -
            :ARG1 (z7 / life
                :ARG1-of (z8 /
                personal-02
                    :ARG2 z4))
            :ARG2 (z9 / mud)
                :purpose (z10 / or
                    :op1 (z11 /
                    score-01
                        :ARG3 (z12 /
                        point)
                        :mod (z13 /
                        politics))
                    :op2 (z14 /
                    consume-01
                        :ARG0 (z15 /
                        public)
                        :ARG1 z7
                        :ARG1-of
                        (z16 /
                        general-02))))
        :ARG1-of (z17 / regardless-91
            :ARG2 z4))
    :mod (z18 / fundamental))
```

manually corrected AMR:

```
""(a / and
    :op1 (e / entitle-01
        :ARG1 (e2 / everyone)
        :ARG2 (l / life
            :ARG1-of (p2 / private-02)))
    :op2 (r / right-05
        :ARG1 e2
        :ARG2 (c / caused-motion-91
            :polarity -
            :ARG1 (l2 / life
                :ARG1-of (p3 / personal-02
                    :ARG2 e2))
            :ARG2 (t / through
                :op1 (m / mud))
            :ARG3 (d / drag-01
                :ARG1 l2)
            :purpose (o / or
                :op1 (s / score-01
```

```
                    :ARG3 (p4 / point)
                    :mod (p5 / politics))
                :op2 (c2 / consume-01
                    :ARG0 (p6 / public)
                    :ARG1 l2
                    :ARG1-of (g / general-02)))
        :ARG1-of (r2 / regardless-91
            :ARG2 e2))
    :mod (f / fundamental))
```

8.1.3. Intransitive

I want you to update the AMR graph to include the new roleset intransitive-motion-91.

Intransitive Motion Construction have a theme that carries out event that causes or accompanies movement in space.

An example sentence with Intransitive Motion Construction:
The fly buzzed into the room

the auto generated graph is:

```
""(z0 / fly-01
    :ARG1-of (z1 / fly-01)
    :destination (z2 / room))""
```

The issue with this is:
Fly.01?? Buzz not represented
It gives us an inaccurate meaning like:"One flying event is the passenger of another flying event into the room."

If we were to fix this to represent the actual meaning we would use Intransitive-Motion.91:
Arg1-theme: The fly
Arg2-goal: into the room
Arg3-concomitant-event: buzz.01
(Arg0: the fly)

Another Common Current AMR Treatment:
Intransitive Motion
Sentence: He ran out of the house.

auto AMR:

```
(z0 / run-02
    :ARG0 (z1 / he)
    :direction (z2 / out-of
        :op1 (z3 / house)))
```

Issue:
Run.02 (Motion semantics align with Intransitive Motion)
ARG0-PPT: runner
ARG1-LOC: course, race, distance
ARG2-PPT: opponent

fix with Intransitive-Motion.91:
Arg1-theme: He
Arg2-goal: out of the house
Arg3-concomitant-event: ran-01
(Arg0: he)

Additional examples:

Sentence: The Weweantic River FLOWS through the pond.

auto-amr:

```
""(z0 / flow-01
    :ARG1 (z1 / river
        :wiki "Weweantic_River"
        :name (z2 / name
            :op1 "Weweantic"
            :op2 "River"))
    :path (z3 / pond))""
```

manually corrected AMR:

```
(i / intransitive-motion-91
    :ARG1 (r / river
        :wiki "Weweantic_River"
        :name (n / name
            :op1 "Weweantic"
            :op2 "River"))
    :ARG2 (p / pond)
    :ARG3 (f / flow-01
        :ARG1 r))
```

Sentence: The river Avon has been strolled along by thousands of famous people throughout history.

auto-amr:

```
(z0 / stroll-01
    :ARG0 (z1 / person
        :ARG1-of (z2 / fame-01)
        :quant (z3 / multiple
            :op1 1000))
    :ARG1 (z4 / river
        :wiki ""Avon_River""
        :name (z5 / name
            :op1 ""Avon""))
    :path (z6 / along)
    :duration (z7 / history))""
```

manually corrected AMR:

```
(i / intransitive-motion-91
    :ARG1 (p / person
        :ARG1-of (f / fame-01)
        :quant (q / multiple
            :op1 1000))
    :ARG2 (a / along
        :op1 (r / river
            :wiki ""Avon_River""
            :name (n / name
            :op1 ""Avon"")))
    :ARG3 (s / stroll-01
        :ARG0 p)
    :duration (h / history))
```

8.1.4. Ditransitive

I want you to update the AMR graph

to include the new roleset ditransitive-91.

Ditransitive Construction is when an Agent carries out an event that is construed as causing a recipient to receive a theme.

An example sentence with Ditransitive Construction
They're going to kill Reagan a commie.

the auto generated graph is:
```
"(z0 / kill-01
    :ARG0 (z1 / they)
    :ARG1 (z2 / person
        :wiki "Ronald_Reagan"
        :name (z3 / name
            :op1 "Reagan")
        :mod (z4 / commie)))"
```

The issue with this is:
Kill.01
ARG0-PAG: killer
ARG1-PPT: corpse
ARG2-MNR: instrument
"They killed Ronald Reagan, who is a commie" is the meaning we get, which is incorrect.

If we were to fix this to represent the actual meaning we would use Ditransitive.91:
Arg0-Cause: They
Arg1-Theme: a commie
Arg2-Recipient: Reagan
Arg3-Concomitant-event: kill

Another Common Current AMR Treatment (ok): Ditransitive

Sentence: Jack poured Jane an arsenic-laced martini.

auto generated AMR:
```
"(z0 / pour-01
    :ARG0 (z1 / person
        :wiki -
        :name (z2 / name
            :op1 "Jack"))
    :ARG1 (z3 / martini
        :ARG1-of (z4 / lace-01
            :ARG2 (z5 / arsenic)))
    :ARG4 (z6 / person
        :wiki -
        :name (z7 / name
            :op1 "Jane")))"
```

The issue:
Pour.01
ARG0-PAG: agent, pourer
ARG1-PPT: liquid
ARG2-DIR: source
ARG3-GOL: destination
"Jack poured an arsenic laced martini (?) Jane" which is not a great interpretation

we can fix this with Ditransitive.91:
Arg0-Cause: Jack
Arg1-Theme: an arsenic laced martini
Arg2-Recipient: Jane
Arg3-Concomitant-event: pour.01
(arg0: Jack, Arg1: martini)

Additional examples:

Sentence: Jack passed her the salt.

auto-amr:
```
"(z0 / pass-05
    :ARG0 (z1 / person
        :wiki -
        :name (z2 / name
            :op1 "Jack"))
    :ARG1 (z3 / salt)
    :ARG2 (z4 / she))"
```

manually corrected AMR:
```
(d / ditransitive-91
    :ARG0 (j / person
        :wiki -
        :name (n / name
            :op1 "Jack"))
    :ARG1 (s / salt)
    :ARG2 (s2 / she)
    :ARG3 (p / pass-05
        :ARG0 j
        :ARG1 s
        :ARG2 s2))
```

Sentence: I no longer think the US Constitution AFFORDS me rights as a citizen .

auto-amr:
```
"(z0 / think-01
    :ARG0 (z1 / i)
    :ARG1 (z2 / afford-02
        :ARG0 (z3 / law
        :wiki
        "United_States_Constitution"
        :name (z4 / name
            :op1 "US"
            :op2 "Constitution"))
        :ARG1 (z5 / right-05
            :ARG1 z1
            :prep-as (z6 / citizen))
        :ARG2 z1)
    :time (z7 / no-longer))"
```

manually corrected AMR:
```
"(t / think-01
```

```
    :ARG0 (i / i)
    :time(n2 / no-longer)
    :ARG1 (d / ditransitive-91
        :ARG0 (l / law
            :wiki
            "United_States_Constitution"
            :name (n / name
                :op1 "US"
                :op2 "Constitution"))
        :ARG1 (r / right-05
            :ARG1 i
            :prep-as (c / citizen))
        :ARG2 i
        :ARG3 (a / afford-02
            :ARG0 l
            :ARG1 r
            :ARG2 i)))"
```

```
Sentence: These skates were bought
for me by my mom.

auto-amr:
"(z0 / buy-01
    :ARG0 (z1 / person
        :ARG0-of (z2 /
        have-rel-role-91
            :ARG1 (z3 / i)
            :ARG2 (z4 / mom)))
    :ARG1 (z5 / skate
         :mod (z6 / this))
    :ARG4 z3)"

manually corrected AMR:
"(d / ditransitive-91
    :ARG0 (m / person
        :ARG0-of (h /
        have-rel-role-91
            :ARG1 (i / me)
            :ARG2 (m2 / mom)))
    :ARG1 (s / skates
        :mod (t / this))
    :ARG2 i
    :ARG3 (b / buy-01
        :ARG0 m
        :ARG1 s
        :ARG4 i))"
```

Appendix C: Cross-Linguistic Validity

Part of the motivation for our work was to capture constructions that occur cross-linguistically. This section will give a small taste of how this might be done for the Cuzco-Collao (QUZ) variety of Quechua, a language with complex, agglutinative morphology.

Quechua is particularly interesting because it has much more explicit Caused Motion constructions than English. In QUZ, the '-chi' affix indicates or delegates action and has an express causative function morphologically, and thus marks a causative construction. Combined in a sentence with the '-ta' object marker on the item being moved, it expresses the meaning that the action is causing something to happen to the patient of the sentence.

(1)

Chaymi	*rumita*	*suchuchirqanku*
Chaymi	rumi-ta	suchu-chi-rqa-nku
therefore	stone-ACC	drag-CAUS-PST-3.PL

aya	*churasqanku*
aya	chura-sqa-nku
corpse	place-Not.Experienced.PST-3PL

t'oqomanta.
t'oqo-manta
hole-ABL.from

'The stone was dragged from the hole where the corpse had been placed.'

We can see in the above example that the causative affix '-chi' attaches to the verb *suchuy* (to drag), while the accusative marker '-ta' attaches to the object *rumi* (stone). This sentence then more literally translates to, 'Therefore they caused the stone to be dragged from the hole where the corpse had been placed (I did not directly see this).'

The basic English representation of this AMR is shown in the following graph.

[1: English AMR] *The stone was dragged from the hole where the corpse had been placed.*

```
(c / caused-motion-91
    :ARG0 (t / they)
    :ARG1 (s / stone)
    :ARG2 (f / from
        :op1 (h / hole
            :location-of (p / place-01
                :ARG1 (c2 / corpse)))))
    :ARG3 (dr / drag-01
            :ARG0 t
            :ARG1 s))
```

Our approach provides a clear way of capturing the meaning in the causative '-chi' morpheme, as seen in the AMR drafted below. It provides a parallel in meaning with the English graph, as the graph structures can remain essentially the same, indicating that these two are semantically similar. The Caused Motion construction in particular would be easy to search for and capture in QUZ. The '-chi' morepheme maps naturally to caused-motion-91 construction. The subject affix on the verb containing '-chi' is then the :ARG0 (in this case *-nku* or 'they'). The :ARG1 is the object that comes before verb with the accusative marker '-ta' (*rumita*). The :ARG2 then can appear in a number of ways, but would generally be any ablative suffix such as '-manta' (*t'oqomanta*).

[2: Quechua AMR] *Chaymi rumita suchuchirqanku aya churasqanku t'oqomanta.*

```
(a / and
    :op1 (c / chi 'caused-motion-91'
        :ARG0 (t / nku 'they')
```

```
        :ARG1 (s / rumi 'stone')
        :ARG2 (f / manta 'from'
            :op1 (h / t'oqo 'hole'
                :location-of (p /
                place-01
                    :ARG1 (c2 /
                    corpse))))
        :ARG3 (d / suchu 'drag-01'
            :ARG0 t
            :ARG1 s))
    :op2 (s2 / sqa 'see-01'
        :polarity -
        :ARG0 (i / i)
        :ARG1 c2
        :manner (d2 / direct-02)))
```

The one addition made to the graph is the evidentiality, which is not present in the plain English graph. The '-sqa' affix is an evidential marker that lets one know that an event was not experienced by the speaker themselves and that the event happened in the past. English does not have grammaticalized evidentiality, and thus it was excluded from the plain English translation graph.

AMR was originally designed for English (and to some extent, Chinese) and therefore does not always directly capture semantic distinctions present in other languages. Thus, the next examples are intended to show how our non-lexically anchored rolesets could be applied beyond AMR to Uniform Meaning Representations (UMR), which is another graph-based semantic framework designed to represent meaning in a cross-linguistically applicable and computationally tractable way as an extension to AMR (Van Gysel et al., 2021).

(2)

Paytaq	*kawsay*	*unuta*
Pay-taq	kawsay	unu-ta
3SG-FOC	life	water-ACC

qosunkiman.
qo-sunki-man
give-3SG>2SG-COND

'He could give you living water.'

The above sentence gives us an example of a Ditransitive sentence in QUZ. The simple English representation of the translation sentence is as follows:

[3: English AMR] *He could give you living water.*

```
(p / possible-01
    :ARG1 d / ditransitive-91
        :ARG0 (h / he)
        :ARG1 (w / water
            :mod living)
        :ARG2 (y / you)
        :ARG3 (g / give-01
                :ARG0 h
                :ARG1 w
                :ARG2 y))
```

Trying to extend this to Quechua, which has a more literal translation of 's/he could give you living water', results in the next graph:

[4: Quechua AMR] *Paytaq kawsay unuta qosunkiman.*

```
(p / possible-01
    :ARG1 d / ditransitive-91
        :ARG0 (p / pay 's/he')
        :ARG1 (u / unu 'water'
            :mod kawsay 'living')
        :ARG2 (n / nki 'you')
        :ARG3 (g / give-01
                :ARG0 p
                :ARG1 u
                :ARG2 n))
```

A notable issue with Graph 4, and Graph 2 for that matter, is that evidentiality is not a concept well captured in AMR representations, but is grammatically expressed in Quechua and, thus, is essential to any accurate semantic representation of its sentences. The `:modstr` value provides a solution to this issue in UMR. Additionally, the idea of genderless 3rd person singular is lost in the English AMR, but recaptured in Graph 5 within UMR. UMR also more accurately depicts verbs in languages that lack a standard PropBank roleset, as seen for the verb 'qoy-00'. For this reason, general CxGr-AMR rolesets would be especially helpful in languages that lack lexical resources or quality mappings to lexical resources in English, as the CxGr-AMR roleset can be used to capture meaning in lieu of costly lexicon development.

[5: Quechua UMR] *Paytaq kawsay unuta qosunkiman.*

```
(d / ditransitive-91
    :ARG0 (p / pay 'person'
        :ref-person 3rd
        :ref-number Singular)
  :ARG1 (u / unu 'water'
        :mod kawsay 'living')
  :ARG2 (s / sunki 'you'
        :ref-person 2nd
        :ref-number Singular)
  :ARG3 (q / qoy-00 'give'
        :actor p
        :theme u
        :recipient s
        :modstr NeutAff
        :aspect Performance))
```

Graph 5 demonstrates that the Ditransitive construction transcends English and can be utilized to accurately depict the semantic relations in QUZ. Our approach could be easily integrated into the UMR schema to more accurately capture cross-linguistic realizations of these types of non-lexically anchored constructions.