Widely Interpretable Semantic Representation: Frameless Meaning Representation for Broader Applicability

Anonymous ACL submission

Abstract

This paper presents a semantic representation called WISeR that overcomes challenges for Abstract Meaning Representation (AMR). Despite its richness and exapandability, AMR is not easily applied to languages or domains without predefined semantic frames, and its use of numbered arguments results in semantic role labels which are not directly interpretable and are semantically overloaded for parsers. We examine the numbered arguments of predicates in AMR and convert them to thematic roles which do not require reference to semantic frames. We create a new corpus of 1K dialogue sentences annotated in both WISeR and AMR. WISeR shows stronger inter-annotator agreement for beginner and experienced annotators, with beginners becoming proficient in WISeR annotation sooner. Finally, we train two state-of-the-art parsers on the AMR 3.0 corpus and a WISeR corpus converted from AMR 3.0. The parsers are evaluated on these corpora and our dialogue corpus. WISeR models exhibit higher accuracy than their AMR counterparts across the board, demonstrating that WISeR is easier for parsers to learn.

1 Introduction

Since Abstract Meaning Representation (AMR; Banarescu et al. (2013)) was introduced, there have been several proposals to extend and/or improve it for deeper and more universal representations (Xue et al., 2019, 2020). This momentum has inspired the development of many parsers (Cai and Lam, 2020; Xu et al., 2020; Lee et al., 2020; Bevilacqua et al., 2021), achieving promising results. A central feature of AMR is its extensive use of PropBank (Palmer et al., 2005; Bonial et al., 2014), which is a corpus of frames that assigns a specific argument structure to every sense of a predicate. Arguments commonly occurring with their predicates are labeled as *numbered arguments* (`ARG`*n*).

There are several advantages of AMR including its simplicity and extendibility. It has a large corpus of annotation (Knight et al., 2014, 2017, 2020), and a significant amount of research has been conducted to enhance AMR's representation of quantifier scope (Pustejovsky et al., 2019; Lai et al., 2020), tense/aspect (Donatelli et al., 2018, 2019), and speech acts (Bonial et al., 2020). Nonetheless, AMR has a few disadvantages. Since AMR largely depends on PropBank to form predicate argument structures, it presupposes the existence of semantic frames for all predicate senses. Consequently, it is not easily adaptable to languages nor to domains in which many new senses appear due to the intense upfront cost in labor to prepare a massive number of frames for novel senses.[1]

Moreover, numbered arguments are semantically opaque without reference to the frames. There is no consistent mapping from numbered arguments to traditional thematic roles which is applicable to all senses besides perhaps `ARG0` and `ARG1`, which correspond to prototypical agent and patient. For instance, `ARG2` of `tell-01` in Figure 1a is the entity which the telling is directed at, while `ARG2` of `dislodge-01` is the initial position of the dislodged entity. Meanwhile, the initial position of the entity stepping-down is the `ARG1` of `step-down-01`. This inconsistent correspondence between numbered arguments and thematic roles makes semantic role labels uninterpretable for parsing models during training. Discussion of these drawbacks is the focus of Section 2.

Section 3 introduces a novel annotation scheme, WISeR (Widely Interpretable Semantic Representation), designed to overcome these challenges. In contrast to AMR, WISeR does not depend on frames. It aims to maintain a one-to-one relation between an argument label and a thematic role, and it has the benefit of permitting the introduction of novel predicates on an *ad-hoc* basis.

[1] Few studies have adapted AMR to other languages (Li et al., 2016; Damonte and Cohen, 2018; Anchiêta and Pardo, 2020; Blloshmi et al., 2020) and domain (Burns et al., 2016).

```
(t / tell-01
   :ARG0 (w / woman)
   :ARG1 (s / step-down-04
            :ARG0 w
            :ARG1 (r / role)
            :time (d / dislodge-01
                     :ARG0 w
                     :ARG1 (b / boss)
                     :ARG2 (b2 / board)))
   :ARG2 (m / man))
```

(a) AMR graph in Penman notation

```
(t / tell
   :actor (w / woman)
   :theme (s / step-down
             :actor w
             :start (r / role)
             :time (d / dislodge
                      :actor w
                      :theme (b / boss)
                      :start (b2 / board)))
   :benefactive (m / man))
```

(b) WISeR graph in Penman notation

Figure 1: AMR and WISeR graphs for the sentence *'The woman told the man she will step down from the role when she dislodges the boss from the board'* in Penman notation (Matthiessen and Bateman, 1991).

Section 4 presents our new corpus comprising 1,000 dialogue sentences annotated in both WISeR and AMR, and makes fair comparisons between the two schemes for annotation adaptability and quality. Section 5 compares parsing models trained on the AMR 3.0 corpus and a WISeR corpus converted from AMR 3.0. Parsing models are evaluated on those corpora as well as our new dialogue corpora, which can be considered an out-of-domain dataset.

To our knowledge, this is the first time that such a large AMR corpus is entirely revised for a "frameless" representation with thematic role labels. We believe this work will facilitate the adaptation of AMR to under-explored domains and languages, thereby building a larger community for meaning representation research.[2]

2 Inside AMR

2.1 Predicates in AMR

AMR annotation begins by identifying disambiguated predicate senses from PropBank frames. Although providing frames as a reference to annotators is designed to ensure consistency during annotation, this disambiguation is often more fine-grained than natural language users are conscious of, leading to low agreement levels in word sense disambiguation tasks (Ng et al., 1999). It also means that AMR is constrained to only a few languages for which frames exist (Palmer et al., 2005; Xue and Palmer, 2005; Palmer et al., 2006; Zaghouani et al., 2010; Vaidya et al., 2011; Duran and Aluísio, 2011; Haverinen et al., 2015; Şahin and Adalı, 2018) and it often lacks domain-specific predicates that occur in certain fields.

AMR contains several predicate senses, however, which are not found in PropBank. These senses often represent idioms or multi-word constructions (e.g., `pack-sand-00`, `throw-under-bus-08`) that are created ad-hoc as the annotation proceeds. Furthermore, there are 9 senses in AMR which have additional numbered arguments not featured in their respective PropBank frames.[3]

	PropBank	AMR 3.0
Total # of predicates	7,311	6,187
Total # of senses	10,687	9,090
Total # of arguments	27,012	23,171
# of unique predicates	1,626	502
# of unique senses	2,153	556

Table 1: Statistics of PropBank and AMR 3.0.

Table 1 shows the statistics of PropBank[4] and the AMR 3.0 release (Knight et al., 2020). We calculate the number of frames in AMR 3.0 by combining information in the release text file[5] with the annotation corpus since there is no subset relation between frames in the text file and those in the corpus, or vice versa. Out of 9,090 senses in AMR 3.0, only 556 are unique to AMR. In other words, 8,534 senses in AMR 3.0 (i.e., 94%) are based on PropBank frames, emphasizing the extent to which AMR annotation depends on PropBank.

2.2 Numbered Arguments in AMR

The argument structure of a predicate sense in PropBank is a set of numbered arguments. As shown in Table 2, the thematic role of benefactive or attribute may be encoded by either `ARG2` or `ARG3`. Consequently, there is no one-to-one correspondence between numbered arguments and thematic

[2]All our resources including the converted WISeR corpus, the new dialogue WISeR corpus, and parsing models are publicly available: https://github.com/anonymous

[3]The 9 senses with additional arguments in AMR: `bind-01: ARG4`, `damage-01: ARG3`, `late-02: ARG3`, `misconduct-01: ARG1`, `oblige-02: ARG2`, `play-11: ARG3`, `raise-02: ARG3`, `rank-01: ARG5`, `unique-01: ARG3-4`

[4]English PropBank frames can be downloaded at https://github.com/propbank/propbank-frames

[5]AMR frames are included in LDC2020T02 as `propbank-amr-frame-arg-descr.txt`

Label	Thematic Role
ARG0	agent
ARG1	patient
ARG2	instrument, benefactive, attribute
ARG3	starting point, benefactive, attribute
ARG4	ending point

Table 2: Numbered arguments and corresponding thematic roles in the PB guidelines (Bonial et al., 2015).

roles. ARG0/ARG1 are intended to correspond to the thematic roles of prototypical agent/patient respectively. However, even this correspondence is occasionally lost. As such, numbered arguments do not directly encode meaning relations. Rather, the semantics of a numbered argument is accessed through two other resources in PropBank: function tags and VerbNet roles (Kipper et al., 2002; Loper et al., 2007). The distribution of function tags over numbered arguments is given in Table 3.[6]

	A0	A1	A2	A3	A4	A5	A6	Σ
PPT	389	8,593	1,249	49	4	0	0	10,284
PAG	8,412	664	28	1	0	0	0	9,105
GOL	2	503	1,436	238	214	2	0	2,395
PRD	0	79	701	231	85	10	0	1,106
MNR	2	10	808	159	8	11	0	998
DIR	18	147	518	270	14	4	0	971
VSP	1	58	338	214	48	19	0	678
LOC	6	196	268	43	25	4	0	542
EXT	1	5	244	25	3	5	6	289
CAU	75	22	140	30	0	0	0	267
COM	0	83	100	9	4	0	0	196
PRP	0	6	74	32	5	1	0	118
TMP	0	3	15	3	6	1	0	28
ADJ	0	5	10	4	0	0	0	19
ADV	0	2	4	5	1	0	0	12
REC	0	1	2	1	0	0	0	4
Σ	8,906	10,377	5,935	1,314	417	57	6	27,012

Table 3: Distribution of function tags (in rows) over numbered arguments (in columns) in PropBank.

This distribution highlights that every numbered argument is semantically opaque without reference to the PropBank frame. As a result, numbered argument role labels make the task of automatic parsing more difficult for machines.

As mentioned, numbered arguments are occasionally annotated with VerbNet roles (Kipper et al., 2008). Unfortunately, the coverage of PropBank frames associated with VerbNet classes is incomplete, with 25.5% of PropBank frames not covered. Even among the PropBank frames which are associated with VerbNet classes there are mismatches; an argument described in one resource may be omitted from the other, or a single argument may be split into multiple arguments. These mismatches reflect both practical and theoretical differences in the resources, and as a result, only 40.6% of arguments in PropBank are mapped to VerbNet roles.[7]

3 Inside WISeR

3.1 Annotation Scheme

This section presents the WISeR annotation scheme, designed to rectify the weaknesses of AMR in Section 2. WISeR does not rely on frames, dispensing with both sense disambiguation and numbered arguments. It represents thematic relations directly as edge labels, similar to the PENMAN Sentence Plan Language (Kasper, 1989) and an earlier version of AMR prior to the incorporation of PropBank (Langkilde and Knight, 1998).

The WISeR graph in Figure 1b above shows how WISeR resolves the issues arising from use of numbered arguments in Figure 1a. Both *role* and *board* stand in the start relation to their predicates In WISeR because they both describe an initial state. However, in AMR, the former is labeled ARG1 and the latter ARG2. Next, both *man* and *board* are labeled as ARG2 in AMR whereas they take distinct thematic roles of *benefactive* and *start* in WISeR. Similarly, the meaning of ARG1 is overloaded in AMR for *role*, *boss*, and *man* as WISeR disambiguates them by assigning the start relation to *role* and theme to *boss* and *man*.

It may seem that the use of thematic roles would lead to a proliferation of semantic relations because there are only a few numbered arguments but many thematic roles. However, this is not the case. WISeR adopts non-core roles that already exist in AMR, allowing annotation of most numbered arguments using these non-core roles. For example, we incorporate the AMR source role with numbered arguments corresponding to initial states into the WISeR start role. We also conflate the beneficiary role in AMR into the WISeR role benefactive, used for annotating thematic benefactive arguments. This reduces redundancy in the annotation scheme since we no longer have two relations fulfilling the same semantic function. We also add a small number of thematic roles based on the PropBank function tags and VerbNet roles. These include the actor and theme roles which broadly correspond to ARG0 and ARG1 in AMR, respectively. The actor role encompasses thematic agent as well as certain non-agentive subjects (e.g.,

[6]The descriptions of these function tag acronyms are provided in Table 12 in Appendix A.1.

[7]The distribution of VerbNet roles over numbered arguments is shown in Table 13 in Appendix A.1.

the bus in *the bus hit the curb*). When all changes are considered, the total number of WISeR roles is fewer than the number of numbered arguments plus non-core roles in AMR. Consequently, WISeR not only reduces the semantic workload of the numbered argument relations, it does so with slightly fewer relations. Finally, WISeR adopts reified relations from AMR such as `have-rel-role` and `have-degree`. The argument structure for each these reified relations is still semi-arbitrary and annotators will need to refer to the guidelines at first.[8]

3.2 Converting AMR to WISeR

To test the relative performance of parsing models on both AMR and WISeR, a mapping is defined to convert all numbered arguments in the AMR 3.0 corpus into WISeR roles. AMR 3.0 is the largest AMR corpus comprising 59,255 sentences collected from various sources including discussion forums, broadcast conversations, weblogs, newswire, children's stories, and more (Knight et al., 2020). There are 556 predicate senses in AMR 3.0 created on an ad-hoc basis (Section 2.1) without reference to a PropBank frame. Sentences which include these ad-hoc senses are removed from this conversion. Furthermore, sentences featuring reified roles with highly specific and non-generalizable argument structures are also removed. For instance, `ARG1-9` of `publication-91` describe *author*, *title*, *abstract*, *text*, *venue*, *issue*, *pages*, *ID*, and *editors*. In total, there are 6 such predicates.[9]

A total of 5,789 predicate senses are collected from PropBank frames that appear at least once in AMR 3.0. The mapping converts every numbered argument for each of these senses to an appropriate WISeR role, totalling 15,120 unique arguments. To define this mapping, the argument number, the function tag, the VerbNet role (if present), and certain keywords in the description are used. The conversion rules and a detailed explanation are presented in Table 17 in Appendix A.2.

The AMR-to-WISeR conversion rules result in a total of 12,311 mappings, which leaves 2,809 numbered arguments in AMR 3.0 that are not automatically mapped to WISeR roles. These are manually mapped using the information in their PropBank frames as well as their specific usage in the corpus. Once all numbered arguments are converted into WISeR roles, sense IDs are removed so that the converted corpus becomes "frameless".

	A0	A1	A2	A3	A4	A5	A6	Σ
THE	57	5,076	256	15	1	0	0	5,405
ACT	4,945	21	9	0	0	0	0	4,975
BEN	1	148	554	90	38	2	0	833
END	0	160	385	51	137	0	0	733
STA	14	63	322	190	6	0	0	595
INS	2	7	441	89	4	3	0	546
ATT	0	6	144	44	6	2	0	202
LOC	1	65	83	7	1	3	0	160
CAU	2	16	115	25	1	0	0	159
PUR	0	11	122	19	5	1	0	158
TOP	2	14	113	20	3	0	0	152
ACC	0	53	69	7	3	0	0	132
OTH	0	21	227	105	15	8	2	378
Σ	5,024	5,661	2,840	662	220	19	2	14,428

Table 4: Distribution of numbered arguments over the most frequent WISeR roles, covering 97.4% of arguments in AMR 3.0. `THE`: theme, `ACT`: actor, `BEN`: benefactive, `END`: end, `STA`: start, `INS`: instrument, `ATT`: attribute, `LOC`: location, `CAU`: cause, `PUR`: purpose, `TOP`: topic, `ACC`: accompanier, `OTH`: other labels.

Table 4 shows the distribution of numbered arguments over the 12 most frequently occurring roles in the converted WISeR corpus. The full version of this table displaying 35 WISeR roles is presented in Table 14 in Section A.1. Although the conversion mappings are created for 15,120 numbered arguments based on the PropBank frames, only 14,428 of them appear in the AMR 3.0 corpus, as shown in the Σ column of the Σ row in Table 4.

4 WISeR Dialogue Corpus

This section presents our new WISeR corpus comprising 1,000 sentences from a variety of dialogue datasets such as EmpatheticDialogues (Rashkin et al., 2018), DailyDialog (Li et al., 2017), Boston English Centre,[10] and PersonaChat (Gu et al., 2020). Additionally, we employ Mechanical Turking tasks to generate 300 sentences, in which subjects are provided with sentences from PersonaChat and asked to respond with emotionally driven reactions (100) or engaging follow-ups (200).

500 of these sentences are evenly split up into 10 batches by making every batch similar in length and complexity. Six batches are split among beginner annotators and are double-annotated in both AMR and WISeR while the other four are divided evenly and double-annotated in either WISeR or AMR by experienced annotators. All annotators are required

[8] The current annotation guidelines for WISeR can be found at our open-source project repository.

[9] The 6 senses with non-generalizable argument structures are: `byline-91`, `course-91`, `distribution-range-91` `publication-91`, `street-address-91`, `statistical-test-91`

[10] 900 English Conversational Sentences from Boston English Centre: https://youtu.be/JP5LYRTZtjw

to annotate in both AMR and WISeR for fair comparison. To control for familiarity, half of the annotators begin in AMR and switch to WISeR while the other half begin in WISeR and switch to AMR.

Beginner annotators are trained for a week and are given additional instructions and feedback with respect to common errors. This is done to minimize orthogonal differences in inter-annotator agreement. The remaining 500 sentences are single-annotated by experienced annotators.

4.1 Inter-Annotator Agreement

To evaluate learnability, inter-annotator agreement (IAA) is estimated by Smatch scores on doubly-annotated batches (Cai and Knight, 2013).

Beginners			Experts		
BID	AMR	WISeR	BID	AMR	WISeR
01	0.72	0.74	07	0.87	-
02	0.72	0.75	08	0.84	-
03	0.68	0.70	09	-	0.89
04	0.69	0.79	10	-	0.85
05	0.77	0.79			
06	0.72	0.76			
μ_b	0.72	**0.76**	μ_e	0.86	**0.87**

Table 5: IAA scores for batches annotated by beginner and expert annotators in AMR and WISeR. BID: batch ID, $\mu_{b/e}$: macro-average scores of the beginner and experienced groups, respectively.

Table 5 shows the IAA scores of individual batches and the macro-average scores of six batches by beginner and four batches by experienced annotators. AMR and WISeR have similar IAA among experts; however, IAA for WISeR is noticeably higher among beginners, implying that AMR has a steeper learning curve, although both schemes produce high-quality annotation once annotators reach the expert-level. All double-annotated sentences are adjudicated with correction.

4.2 Annotation Time

Every beginner annotator is assigned 3 batches and asked to report annotation times for each batch, allowing us to compare how quickly they become proficient in annotating either scheme. These results are summarized in Table 6. For Batches 1 and 2 there is practically no difference in time between AMR and WISeR annotation. However, for Batch 3, annotating in WISeR is quicker. This is likely due to familiarization with the WISeR guidelines and experience choosing the appropriate WISeR roles, while the process of identifying the correct frames and numbered arguments in AMR remains the same regardless of experience.

AID	AMR			WISeR		
	1	2	3	1	2	3
A	115	123	121	114	112	114
B	66	67	67	66	67	66
C	129	87	95	105	91	94
D	106	138	128	124	144	138
E	154	131	127	146	93	78
F	122	75	-	140	105	-
μ_a	**115**	104	107	116	**102**	**98**

Table 6: Time it takes for each of 6 annotators to annotate 3 batches. Annotator F completed only the first two batches. AID: annotator ID.

4.3 Corpus Analytics

Table 7 shows the statistics of our dialogue corpus annotated in AMR and WISeR, providing diverse utterances from six sources. DailyDialog, Boston English Center, and EmpatheticDialogues have longer utterances as they are commonly in narrative form. PersonaChat consists of slightly shorter utterances, but its structures are still relatively complex. Utterances in MTurk-Followup are mostly interrogatives and are shorter than ones from the other three. MTurk-Reaction utterances are the shortest since they are mainly emotional reactions (e.g., *that's impressive*). These six sources yield 8.3K+ tokens with 5.4K+ concepts and 5.2K+ relations, allowing researchers to make meaningful parsing evaluation on the dialogue domain.[11]

In comparison, the Dialogue-AMR corpus (Bonial et al., 2020) consists of 80 hours of commands and requests made by humans to robots in search and navigation tasks. It is mostly limited to these specific speech acts and mainly focuses on spatial words. Our dialogue corpus, on the other hand, contains personal interactions about the speakers' likes and dislikes, relationships, and day-to-day life, aimed at creating a personal and meaningful relationship with their interlocutor. Our corpus is also publicly available whereas no public access is currently available for the Dialogue-ARM corpus.

5 Experiments

To assess the interpretability of the WISeR scheme, two state-of-the-art parsers (Sections 5.2 and 5.3) are trained and tested on trimmed AMR 3.0 (AMR_t)[12] and the WISeR corpus converted from

[11] At present, our corpus does not feature Wikification. However, we intend to include this in a near future release.

[12] Sentences including ad-hoc predicates are removed in the trimmed AMR_t corpus as described in Section 3.2.

Source	Sent.	Tokens	Concepts		Relations		Reent.		Negations		NE	
			A	W	A	W	A	W	A	W	A	W
DailyDialog	200	2,177	1,297	1,298	1,315	1,318	211	229	27	26	21	22
Boston English Center	200	1,989	1,182	1,196	1,167	1,179	217	219	33	33	12	13
PersonaChat	200	1,431	962	961	921	911	147	153	18	17	32	30
EmpatheticDialogues	100	1,090	692	699	712	710	131	128	20	20	1	1
MTurk-Followup	200	1,368	1,037	1,040	935	928	134	137	7	7	10	8
MTurk-Reaction	100	298	260	256	191	180	14	15	7	6	0	0
Σ	1,000	8,353	5,433	5,447	5,240	5,226	854	881	112	109	76	74

Table 7: Statistics of our dialogue corpus (in counts) by different categories annotated in AMR (A) and WISeR (W). Sent: sentences, Reent: Reentrancies, NE: named entities.

AMR3$_t$ (WISeR$_c$). The AMR$_t$ parsing models are additionally tested on our dialogue corpus annotated in AMR (ADC). Finally, the WISeR$_t$ models are evaluated on the ADC converted into WISeR (WDC$_c$), maintaining consistency with WISeR$_c$, as well as our dialogue corpus manually annotated in WISeR (WDC$_m$). The key differences between WDC$_c$ and WDC$_m$ are discussed in Section 5.6.

5.1 Datasets

Table 8 shows the number of sentences in each split for the datasets used in our experiments.

Set	AMR 3.0	AMR$_t$ | WISeR$_c$	ADC | WDC$_{c\|m}$
`TRN`	55,635	53,296	-
`DEV`	1,722	1,656	-
`TST`	1,898	1,813	1,000
Σ	59,255	56,765	1,000

Table 8: Number of sentences in the training (`TRN`), development (`DEV`), and evaluation (`TST`) sets.

ADC and WDC$_{c|m}$ are annotations of the same dialogue corpus and are used only for evaluation. In the future, we plan to create a larger corpus of manual WISeR annotations to train more robust parsers for the dialogue domain.

5.2 Graph-based Parser

We first adopt a graph-sequence iterative parser by Cai and Lam (2020) that incrementally builds an AMR graph by expanding one concept at a time. Taking a sentence and a partial graph as input, it uses two transformers to create token and concept embeddings, respectively. These embeddings are fed into paired transformer layers for arc prediction and representation learning. The next concept embedding created by these layers is fed to another arc generation layer, which initiates another round of iteration. Once the iterative inference is finished, the final concept embeddings are decoded into concepts through beam search and arcs between these concepts are predicted by another arc generation layer. Finally, the arc labels are predicted by a biaffine layer taking the concept embeddings as input (Dozat and Manning, 2017).

5.3 Seq-to-Seq Parser

We also adopt a seq-to-seq parser, SPRING, which currently holds the highest parsing accuracy on AMR 3.0 (Bevilacqua et al., 2021). SPRING linearizes every graph into a sequence of tokens in the depth-first search order and trains the sequence using a seq-to-seq model called BART (Lewis et al., 2020). In this sequence, special tokens are used to indicate variables and parentheses in the PENMAN notation. Given a sentence and its linearized graph, BART is finetuned to learn the transduction from the former to the latter. Once a linearized graph is generated, parenthesis parity is restored and any token that is not a possible continuation given the previous token is removed. In our experiments, the BART large model with greedy decoding is used.

5.4 Parsing Results

Table 9 shows the performance of the graph-based parser and the seq-to-seq parser on the five datasets, with Smatch scores (Cai and Knight, 2013), as well as more fine-grained metrics (Damonte et al., 2017). Comparing the results on AMR$_t$ and WISeR$_c$, the WISeR parsers outperform the AMR parsers on all categories, showing $\approx$1% higher Smatch scores for both parsers, which implies that WISeR is easier to learn, enabling these parsers to train more robust models. The *No WSD* (no word sense disambiguation) scores for WISeR are equivalent to the Smatch scores because predicates in WISeR are not distinguished by senses. Unsurprisingly, the WISeR parsers show higher scores on this category confirming that WSD introduces an extra burden on the AMR parsers. For *Concepts* and *Negations*, the WISeR parsers also show significant improvement over the AMR parsers; $\approx$3% and 6%, respectively. The *SRL* (semantic role labeling) metric is only de-

Dataset	Smatch	Unlabeled	No WSD	Concepts	xSRL	Reentrancies	Negations	Named Entity
AMR_t	77.2 ± 0.1	80.4 ± 0.2	77.7 ± 0.2	86.6 ± 0.1	68.4 ± 0.2	63.3 ± 0.2	73.0 ± 0.2	73.6 ± 0.6
WISeR_c	$\mathbf{78.5 \pm 0.1}$	$\mathbf{81.5 \pm 0.1}$	$\mathbf{78.5 \pm 0.1}$	$\mathbf{89.4 \pm 0.2}$	$\mathbf{68.9 \pm 0.2}$	$\mathbf{64.1 \pm 0.1}$	$\mathbf{78.9 \pm 0.4}$	$\mathbf{74.0 \pm 0.4}$
ADC	76.7 ± 0.3	81.1 ± 0.3	77.9 ± 0.4	85.0 ± 0.2	75.8 ± 0.0	69.0 ± 0.7	63.8 ± 1.3	36.0 ± 2.1
WDC_c	$\mathbf{79.0 \pm 0.1}$	81.9 ± 2.6	$\mathbf{79.0 \pm 0.1}$	88.6 ± 0.2	$\mathbf{76.6 \pm 0.2}$	$\mathbf{69.9 \pm 0.3}$	$\mathbf{70.7 \pm 0.9}$	$\mathbf{39.6 \pm 4.3}$
WDC_m	78.2 ± 0.2	$\mathbf{83.3 \pm 0.1}$	78.2 ± 0.2	$\mathbf{88.6 \pm 0.1}$	73.7 ± 0.5	68.4 ± 0.4	70.4 ± 1.0	38.4 ± 3.8

(a) Parsing performance achieved by the graph-based models in Section 5.2.

Dataset	Smatch	Unlabeled	No WSD	Concepts	xSRL	Reentrancies	Negations	Named Entity
AMR_t	83.5 ± 0.1	85.9 ± 0.0	84.0 ± 0.1	90.3 ± 0.0	75.9 ± 0.2	71.4 ± 0.3	73.0 ± 1.0	88.7 ± 0.5
WISeR_c	$\mathbf{84.4 \pm 0.1}$	$\mathbf{86.7 \pm 0.1}$	$\mathbf{84.4 \pm 0.1}$	$\mathbf{93.0 \pm 0.1}$	$\mathbf{76.2 \pm 0.4}$	$\mathbf{71.9 \pm 0.2}$	$\mathbf{78.9 \pm 0.2}$	$\mathbf{88.7 \pm 0.4}$
ADC	80.3 ± 0.2	83.8 ± 0.1	81.4 ± 0.2	86.8 ± 0.0	78.8 ± 0.3	71.8 ± 0.8	70.3 ± 0.5	65.5 ± 1.4
WDC_c	$\mathbf{82.3 \pm 0.2}$	85.7 ± 0.2	$\mathbf{82.3 \pm 0.2}$	90.8 ± 0.1	$\mathbf{79.2 \pm 0.3}$	$\mathbf{72.8 \pm 0.3}$	76.2 ± 0.9	68.2 ± 1.8
WDC_m	81.5 ± 0.2	$\mathbf{85.9 \pm 0.2}$	81.5 ± 0.2	$\mathbf{91.1 \pm 0.1}$	75.9 ± 0.2	70.6 ± 0.4	$\mathbf{78.2 \pm 0.1}$	$\mathbf{74.9 \pm 1.0}$

(b) Parsing performance achieved by the seq-to-seq models in Section 5.3.

Table 9: Performance of the graph-based parser and the seq-to-seq parser on the five evaluation sets.

fined for numbered arguments and so is not applicable to WISeR. To assess core argument labeling in both schemes, we propose a new metric called *xSRL* (extended SRL). The xSRL metric compares the WISeR roles in Table 4 against `ARG0-6` plus a few non-core roles in AMR, which correspond to the WISeR roles in Table 4.[13] The WISeR parsers again outperform the AMR parsers in this category.

Comparing the results on the ADC and WDC_c, which are out-of-domain datasets, we find the same trend. The performance gain here is even larger as the WISeR parsers produce Smatch scores higher by $\approx 2\%$. This indicates that the WISeR parsers handle the dialogue domain better. Surprisingly, scores on the dialogue corpus are higher for *xSRL* and *Reentrancies* for all parsing models than ones on AMR_t and WISeR_c. This may be due to smaller graphs and possibly simpler argument structures in the dialogue corpus.[14]

Comparing the results of WDC_c and WDC_m, it is expected that WDC_c should score better than WDC_m due to discrepancies between converted and manual annotation. However, the unlabeled scores are slightly higher on WDC_m for both parsers, implying that the WISeR models still find the correct representations for out-of-domain data. The named entity results of the seq-to-seq model are 6.5% higher on WDC_m than WDC_c which is encouraging for areas such as Conversational AI that rely heavily on named entity recognition.

5.5 Error Analysis

For the graph-based parsers, WISeR relations provide more consistent teaching signals than the often overloaded semantic roles (Section 2.2), which ultimately improve the representation of concepts. In addition, the seq-to-seq parsers also benefit from the more natural relation names in WISeR which are learnt during the pre-training of BART.

The WISeR parser has the freedom to coin novel concepts for predicate senses on which it lacks sufficient training. For example, the verb *premeditate* is absent from the training data, but present in the test set of AMR_t and WISeR_c. Out of 3 runs, the seq-to-seq AMR parser predicts the correct concept `premeditate-01` only once, predicting the concept `intend-01` once and `deliberate-01` once. In comparison, the seq-to-seq WISeR parser uses the novel concept `premeditate` every time. The set of frames that occur only in the test set is rather small, so to make a fair comparison when evaluating the performance on the AMR_t corpus, we restrict our comparison to the subset of novel frames which do not correspond to concepts in the WISeR_c training data after conversion.[15] When comparing on the dialogue corpus, we restrict our comparison to those concepts which are annotated identically in WDC_m and WDC_c, and the concepts in AMR which feed into WDC_c. We thus compare performance only on words which are translated into a novel predicate concept in every dataset. The recall of the seq-to-seq parser across the evaluation sets is shown in Table 10.

Finally, we tested the seq-to-seq parser on the

[13]The non-core roles are: `accompanier`, `beneficiary`, `destination`, `instrument`, `location`, `purpose`, `source`, and `topic`. The AMR role `cause` is not used in the AMR 3.0 corpus.

[14]Our experimental settings are provided in Appendix A.3.

[15]E.g., `move-04` is absent in the AMR training set but present in the test set. It is not included in the comparison since it is converted to `move` which is present in the WISeR training.

Dataset	Recall	Dataset	Recall
AMR_t	0.57	ADC	0.28
$WISeR_c$	0.80	WDC_c	0.42
		WDC_m	0.60

Table 10: Recall of the seq-to-seq parser on novel predicate concepts in the five evaluation sets.

WSD and SRL tasks independently. The bottom left cell in Table 11 is the Smatch score for the WISeR parser, and the top right is the AMR parser. The top left is a parser trained with PropBank senses and automatically converted WISeR roles, while the bottom right used numbered ARGs without predicate senses.[16]

	WISeR roles	Numbered ARGs
+WSD	83.8 ± 0.1	83.5 ± 0.1
-WSD	84.4 ± 0.1	84.2 ± 0.1

Table 11: Comparing the effect of transparent SRL and removing WSD independently.

This shows a $\approx$0.3% increase when using WISeR roles over numbered arguments even with predicate senses, while removing predicate senses accounts for a larger $\approx$0.7% increase.

5.6 Challenges

A potential challenge in these experiments is that the converted WISeR corpus, $WISeR_c$, is arguably only pseudo-WISeR. For instance, many predicate concepts corresponding to adjectives (e.g., *great*) do not have PropBank frames. Consequently, the sentence *that is great* is annotated using the role `domain` in AMR but `theme` in WISeR. Such inconsistency introduces noise to parsing models that leads to suboptimal performance.

For our dialogue corpus, the difference between the manual WISeR annotation, WDC_m, and the converted WISeR annotation, WDC_c, is quantified by running the Smatch metric on those two sets. A Smatch score of 0.88 is returned for this comparison. Although relatively high, this does indicate a training-evaluation discrepancy. Besides the unavailability of certain PropBank frames, this could also be partially due to different annotators. In the near future, we plan to enhance the automatic conversion to close down this gap as much as possible.

5.7 Discussions

A potential explanation for why the WISeR parser outperforms the AMR parser is that many WISeR roles are associated with surface level syntax in the object language. For example, a `topic` argument is often introduced with the preposition *about* or *on*, an `end` is typically introduced by the preposition *to*, `start` with *from* or *out of* etc. These cues are obscured when a single numbered argument encodes more than one thematic role, or when one thematic role is encoded by more than one numbered argument. In WISeR, however, there is a one-to-one correspondence between any relation (edge label), and its semantic function (thematic role). As such, syntactic cues indicating the appropriate WISeR role can be found in the data, making classification easier and increasing parser accuracy. Moreover, assigning consistent, more meaningful labels can help with data sparsity, while also capitalizing on the understanding that pre-trained models already have of the language.

Finally, since automatically converted WISeR roles can be used with PropBank predicate senses, researchers can still make use of PropBank resources if they are required for inference tasks later down the line, while nonetheless employing more transparent semantic role labels during parsing, albeit with more modest improvements.

6 Conclusion

AMR relies on PropBank frames to disambiguate predicate senses and provide a predefined argument structure for each of these senses. This paper discusses several downsides of this approach. Due to the absence of appropriate frames, AMR is currently limited to a handful of languages. Also, numbered arguments in PropBank are semantically opaque, as each role (even `ARG0` and `ARG1`) encodes multiple thematic roles across frames.

In a bid to rectify these problems, this paper introduces a novel annotation scheme, WISeR. Our findings show that WISeR supports improved parsing performance as well as annotation of equal (or better) quality in less time. Based on these results, we conclude that the removal of numbered arguments and sense disambiguation in favor of thematic roles alleviates potential issues associated with AMR's use of PropBank frames, making WISeR easier to learn for parsers.

We will continue to explore new methods of improving WISeR and increase the size of our corpus in volume as well as diversity for other languages so that WISeR parsing models can be robust enough to be broadly used in practice.

[16] Since the use of numbered arguments depends on the sense-disambiguation of predicates, WSD and SRL tasks are not sensibly separated if using numbered arguments.

References

Rafael Anchiêta and Thiago Pardo. 2020. Semantically inspired AMR alignment for the Portuguese language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1595–1600, Online. Association for Computational Linguistics.

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

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*, AAAI'21.

Rexhina Blloshmi, Rocco Tripodi, and Roberto Navigli. 2020. XL-AMR: Enabling cross-lingual AMR parsing with transfer learning techniques. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2487–2500, Online. Association for Computational Linguistics.

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D. Hwang, and Martha Palmer. 2014. PropBank: Semantics of new predicate types. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3013–3019, Reykjavik, Iceland. European Language Resources Association (ELRA).

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D. Hwang, Martha Palmer, and Nicholas Reese. 2015. English PropBank Annotation Guidelines.

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

Gully A. Burns, Ulf Hermjakob, and José Luis Ambite. 2016. Abstract Meaning Representations as Linked Data. In *Proceedings of the International Semantic Web Conference*, ISWC'16, pages 12–20.

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.

Marco Damonte and Shay B. Cohen. 2018. Cross-lingual Abstract Meaning Representation parsing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1146–1155, New Orleans, Louisiana. Association for Computational Linguistics.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for Abstract Meaning Representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546, Valencia, Spain. Association for Computational Linguistics.

Lucia Donatelli, Michael Regan, William Croft, and Nathan Schneider. 2018. Annotation of tense and aspect semantics for sentential AMR. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 96–108, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Lucia Donatelli, Nathan Schneider, William Croft, and Michael Regan. 2019. Tense and aspect semantics for sentential AMR. *Proceedings of the Society for Computation in Linguistics*, 2(1):346–348.

Timothy Dozat and Christopher D. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR'17.

Magali Sanches Duran and Sandra Maria Aluísio. 2011. Propbank-br: a Brazilian Portuguese corpus annotated with semantic role labels. In *Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology*.

Jia-Chen Gu, Zhen-Hua Ling, Xiaodan Zhu, and Quan Liu. 2020. Dually Interactive Matching Network for Personalized Response Selection in Retrieval-Based Chatbots.

Katri Haverinen, Jenna Kanerva, Samuel Kohonen, Anna Missilä, Stina Ojala, Timo Viljanen, Veronika Laippala, and Filip Ginter. 2015. The Finnish Proposition Bank. *Language Resources and Evaluation*, 49(4):907–926.

Robert T. Kasper. 1989. A flexible interface for linking applications to Penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A Large-scale Classification of English Verbs. *Language Resources and Evaluation*, 42(1):21–40.

Karin Kipper, Martha Palmer, and Owen Rambow. 2002. Extending PropBank with VerbNet Semantic Predicates. In *Proceedings of the AMTA Workshop on Applied Interlinguas*.

Kevin Knight, Bianca Badarau, Laura Banarescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and Nathan Schneider. 2017. Abstract Meaning Representation (AMR) Annotation Release 2.0.

Kevin Knight, Bianca Badarau, Laura Banarescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and Nathan Schneider. 2020. Abstract Meaning Representation (AMR) Annotation Release 3.0.

Kevin Knight, Laura Banarescu, Claire Bonial, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, and Nathan Schneider. 2014. Abstract Meaning Representation (AMR) Annotation Release 1.0.

Kenneth Lai, Lucia Donatelli, and James Pustejovsky. 2020. A continuation semantics for Abstract Meaning Representation. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 1–12, Barcelona Spain (online). Association for Computational Linguistics.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 704–710, Montreal, Quebec, Canada. Association for Computational Linguistics.

Young-Suk Lee, Ramón Fernandez Astudillo, Tahira Naseem, Revanth Gangi Reddy, Radu Florian, and Salim Roukos. 2020. Pushing the limits of AMR parsing with self-learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3208–3214, Online. Association for Computational Linguistics.

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.

Bin Li, Yuan Wen, Weiguang Qu, Lijun Bu, and Nianwen Xue. 2016. Annotating the little prince with Chinese AMRs. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 7–15, Berlin, Germany. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. *CoRR*, abs/1710.03957.

Edward Loper, Szu ting Yi, and Martha Palmer. 2007. Combining lexical resources: Mapping between PropBank and VerbNet. In *In Proceedings of the 7th International Workshop on Computational Linguistics*.

Christian M. I. M. Matthiessen and John A Bateman. 1991. *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Pinter.

Hwee Tou Ng, Chung Yong Lim, and Shou King Foo. 1999. A case study on inter-annotator agreement for word sense disambiguation. In *SIGLEX99: Standardizing Lexical Resources*.

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, Shijong Ryu, Jinyoung Choi, Sinwon Yoon, and Yeongmi Jeon. 2006. Korean Propbank.

James Pustejovsky, Ken Lai, and Nianwen Xue. 2019. Modeling quantification and scope in Abstract Meaning Representations. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 28–33, Florence, Italy. Association for Computational Linguistics.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2018. I Know the Feeling: Learning to Converse with Empathy. *CoRR*, abs/1811.00207.

Gözde Gül Şahin and Eşref Adalı. 2018. Annotation of semantic roles for the Turkish Proposition Bank. *Language Resources and Evaluation*, 52(3):673–706.

Ashwini Vaidya, Jinho Choi, Martha Palmer, and Bhuvana Narasimhan. 2011. Analysis of the Hindi Proposition Bank using dependency structure. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 21–29, Portland, Oregon, USA. Association for Computational Linguistics.

Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2020. Improving AMR parsing with sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2501–2511, Online. Association for Computational Linguistics.

Nianwen Xue, Johan Bos, William Croft, Jan Hajič, Chu-Ren Huang, Stephan Oepen, Martha Palmer, and James Pustejovsky, editors. 2020. *Proceedings of the Second International Workshop on Designing Meaning Representations*. Association for Computational Linguistics, Barcelona Spain (online).

Nianwen Xue, William Croft, Jan Hajic, Chu-Ren Huang, Stephan Oepen, Martha Palmer, and James Pustejovksy, editors. 2019. *Proceedings of the First International Workshop on Designing Meaning Representations*. Association for Computational Linguistics, Florence, Italy.

Nianwen Xue and Martha Palmer. 2005. Automatic Semantic Role Labeling for Chinese Verbs. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 1160–1165.

Wajdi Zaghouani, Mona Diab, Aous Mansouri, Sameer Pradhan, and Martha Palmer. 2010. The revised Arabic PropBank. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 222–226, Uppsala, Sweden. Association for Computational Linguistics.

A Appendix

A.1 Argument Descriptions

Table 12 shows function tags used to disambiguate fine-grained roles of numbered arguments in PropBank frames.

Tag	Description	Tag	Description
PPT	Prototypical Patient	**EXT**	Extent
PAG	Prototypical Agent	**CAU**	Cause
GOL	Goal	**COM**	Comitative
PRD	Secondary Predication	**PRP**	Purpose
MNR	Manner	**TMP**	Temporal
DIR	Directional	**ADJ**	Adjectival
VSP	Verb-specific	**ADV**	Adverbial
LOC	Locative	**REC**	Reciprocal

Table 12: Descriptions of the function tags in PropBank.

Table 13 shows the distribution of VerbNet thematic roles (in rows) over the numbered arguments (in columns) in PropBank frames. Not all numbered arguments in the PropBank frames are aligned with VerbNet roles as only 40.6% of arguments in these frames are mapped to specific VerbNet roles.

Table 14 shows the distribution of WISeR thematic roles (in rows) over the numbered arguments (in columns) in PropBank frames, which is the full version of Table 4 in Section 3.2.

A.2 AMR-to-WISeR Conversion

The conversion rules in Table 17 are used to convert numbered arguments into WISeR roles. Two or more of the following sources of information in PropBank are used to compute a conversion: the number of the argument, the functional tag, the VerbNet role (if present), and an informal description of the argument written by PropBank annotators. For example, if an instance of an `ARG1` is labeled with a `PAG` function tag in PropBank and has a description containing either "entity" or "thing", then it is mapped to the WISeR role `theme` (see row 4 of Table 17). Using these mappings, for each AMR graph, all numbered argument edge labels were identified and relabeled with their WISeR role. We also relabeled AMR non-core roles of `source` to WISeR `start`, `destination` to WISeR `end`, `beneficiary` to WISeR `benefactive`, and `medium` to WISeR `manner`. Lastly, we converted concepts like `amr-unknown` and `amr-choice` into their WISeR counterparts.

A.3 Experimental Settings

The hyper-parameter settings for the graph parser (Section 5.2) and the seq2seq parser (Section 5.3) are described in Table 16 and 15, respectively.

	ARG0	ARG1	ARG2	ARG3	ARG4	ARG5	Σ
agent	3,462	30	1	1	0	0	3,494
theme	208	1,661	371	13	0	0	2,253
patient	13	1,131	20	0	0	0	1,164
experiencer	187	264	5	2	0	0	458
destination	0	231	183	21	10	1	446
stimulus	247	172	14	0	0	0	433
location	7	145	142	30	23	1	348
source	17	109	194	7	2	0	329
recipient	0	56	251	10	0	0	317
instrument	0	2	243	51	0	3	299
topic	0	192	61	5	0	0	258
co-patient	0	6	151	4	1	0	162
beneficiary	0	40	47	44	7	0	138
attribute	0	9	101	7	2	6	125
result	0	30	81	5	7	0	123
co-agent	0	69	25	0	0	0	94
material	1	25	46	9	0	0	81
goal	0	8	58	6	1	0	73
co-theme	0	37	27	5	1	0	70
product	0	35	17	4	13	0	69
initial_location	0	9	23	8	0	0	40
cause	30	3	3	0	0	0	36
asset	0	21	0	11	1	1	34
predicate	0	4	18	6	0	0	28
pivot	26	1	0	0	0	0	27
extent	0	0	26	6	0	0	26
value	0	5	13	7	0	0	25
trajectory	0	3	0	0	0	0	3
actor	1	0	0	0	0	0	1
proposition	0	0	0	1	0	0	1
Σ	4,199	4,298	2,121	257	68	12	10,955

Table 13: Distribution of VerbNet thematic roles over numbered arguments in PropBank.

	ARG0	ARG1	ARG2	ARG3	ARG4	ARG5	ARG6	Σ
theme	57	5,076	256	15	1	0	0	5,405
actor	4,945	21	9	0	0	0	0	4,975
benefactive	1	148	554	90	38	2	0	833
end	0	160	385	51	137	0	0	733
start	14	63	322	190	6	0	0	595
instrument	2	7	441	89	4	3	0	546
attribute	0	6	144	44	6	2	0	202
location	1	65	83	7	1	3	0	160
cause	2	16	115	25	1	0	0	159
purpose	0	11	122	19	5	1	0	158
topic	2	14	113	20	3	0	0	152
accompanier	0	53	69	7	3	0	0	132
extent	0	0	77	8	2	0	0	87
comparison	0	1	51	7	3	3	2	67
asset	0	1	11	53	1	0	0	66
domain	0	4	23	11	0	0	0	38
mod	0	2	15	4	1	0	0	22
manner	0	3	9	5	2	0	0	19
direction	0	0	7	0	2	5	0	14
path	0	7	4	1	0	0	0	12
cause-of	0	0	6	2	1	0	0	9
degree	0	0	3	5	1	0	0	9
subevent	0	0	3	2	1	0	0	6
quantity	0	1	4	0	0	0	0	5
value	0	0	3	2	0	0	0	5
time	0	1	2	1	0	0	0	4
part-of	0	1	1	2	0	0	0	4
duration	0	0	2	0	1	0	0	3
theme-of	0	0	2	0	0	0	0	2
range	0	0	1	0	0	0	0	1
poss	0	0	1	0	0	0	0	1
example	0	0	0	1	0	0	0	1
consist-of	0	0	1	0	0	0	0	1
concession	0	0	1	0	0	0	0	1
frequency	0	0	0	1	0	0	0	1
Σ	5,024	5,661	2,840	662	220	19	2	14,428

Table 14: Distribution of PropBank numbered arguments to WISeR thematic roles.

BART	
version	large
# parameters	406M
layers	24
hidden size	1024
heads	16
Adam Optimizer	
learning rate	5e-5
warm up steps	0
weight decay	0.004
batch #tokens	5000
epochs	30

Table 15: Hyper-parameters for the seq2seq parser.

Embeddings	
lemma	300
POS tag	32
NER tag	16
concept	300
char	32
Char-level CNN	
#filters	256
ngram filter size	3
output size	128
Text Encoder	
#transformer layers	4
Graph Encoder	
#transformer layers	2
Transformer Layer	
#heads	8
hidden size	512
feed-forward hidden size	1024
Graph Transformer	
feed-forward hidden size	1024
Biaffine	
hidden size	100

Table 16: Hyper-parameters for the graph parser.

ARGx	F-Tag	VerbNet Role	Description	WISeR Role
+ARG0	+PAG			Actor
+ARG0	+CAU			Actor
+ARG1	+PPT			Theme
+ARG1	+PAG		+(entity|thing)	Theme
	+MNR	+instrument		Instrument
	+MNR	-instrument		Manner
	+GOL	+destination		End
	+GOL		+ (end point|ending point| state|destination|attach| attached|target)	End
	+GOL	+ (beneficiary|recipient| experiencer)		Benefactive
	+GOL		(benefactive|beneficiary|recipient| listener|hearer|perceiver|to whom| pay|paid)	Benefactive
	+LOC	+destination		End
	+LOC	+initial_location		Start
	+LOC	+source		Start
	+LOC	-destination		Location
	+LOC		+(end point|ending point|state| destination|attach|target|end)	End
	+LOC		+(start|source|from|starting)	Start
	+DIR	+initial_location		Start
	+DIR	+source		Start
	+DIR		+(start|source|from|starting)	Start
	+COM	-recipient & -beneficiary		Accompanier
	+COM	+(recipient|beneficiary)		Benefactive
+ARG1	+VSP	+asset		Theme
	+VSP		+(price|money|rent| amount|gratuity)	Asset
	+PRP		+(purpose|for)	Purpose
-ARG1	+CAU	-recipient	+(why|reason|source| cause|crime|because)	Cause
	+VSP	+(material|source)		Start
	+VSP		+(start|material|source)	Start
	+VSP		+(aspect|domain) & -specific	Domain

Table 17: WISeR role mappings from ARGx, f-tag, VerbNet role, and description information.