The Role of PropBank Sense IDs in AMR-to-text Generation and Text-to-AMR Parsing

Thu Hoang
Amherst College
thuhoang28@amherst.edu

Mina Yang
Amherst College
miyang27@amherst.edu

Shira Wein
Amherst College
swein@amherst.edu

Abstract

The graph-based semantic representation Abstract Meaning Representation (AMR) incorporates Proposition Bank (PropBank) sense IDs to indicate the senses of nodes in the graph and specify their associated arguments. While this contributes to the semantic information captured in an AMR graph, the utility of incorporating sense IDs into AMR graphs has not been analyzed from a technological perspective, i.e. how useful sense IDs are to generating text from AMRs and how accurately senses are induced by AMR parsers. In this work, we examine the effects of altering or removing the sense IDs in the AMR graphs, by perturbing the sense data passed to AMR-to-text generation models. Additionally, for text-to-AMR parsing, we quantitatively and qualitatively verify the accuracy of sense IDs produced from state-of-the-art models. Our investigation reveals that sense IDs do contribute a small amount to accurate AMR-to-text generation, meaning they enhance AMR technologies, but may be disregarded when their reliance prohibits multilingual corpus development.

1 Introduction

The Proposition Bank (PropBank; Palmer et al., 2005) is a corpus of semantic roles of verbs and their arguments, where each verb sense is assigned an ID.[1] In addition to verbs, PropBank also annotates semantic roles of select adjectives, prepositions, and multiword expressions (Pradhan et al., 2022); some of the verbs in PropBank are verb-particle constructions, where a combination of a verb and preposition have a specified unique meaning, such as “turn in” meaning to submit/hand in.

Text: Everyone likes strawberries in summer.
Parsed AMR:

```
(l / like-01
    :ARG0 (e / everyone)
    :ARG1 (s / strawberry)
    :time (s2 / summer))
```

Figure 1: Example sentence and its AMR graph. The dashed number (`-01`) is the PropBank sense ID specifying the intended meaning of the predicate `like`.

The graph-based semantic representation Abstract Meaning Representation (AMR; Banarescu et al., 2013) uses English PropBank frames to indicate the sense of each node in the graph and its associated arguments (as shown for `like-01` in Figure 1). While the sense IDs in AMR graphs provide relevant semantic information, this inclusion requires manually checking PropBank for each sense ID and presents challenges when trying to annotate AMR in languages other than English (if adequate PropBank frames do not exist for that language). Senses do not always correspond across languages (Padó, 2007; van der Plas et al., 2010), limiting the benefits of relying on English PropBank for non-English languages, and many low-resource languages do not have framesets available. Two extensions of AMR, Uniform Meaning Representation (UMR; Van Gysel et al., 2021) and WISeR (Widely Interpretable Semantic Representation; Feng et al., 2023), resolve this issue by incorporating a “Stage 0” frameset development phase for low-resource languages and eliminating senses from the representation entirely, respectively.

Thus, given the prohibitive nature of sense IDs in multilingual extensions of AMR, in this work, we examine the technical utility of maintaining sense IDs in AMRs. We investigate the extent to which AMR-to-text generation models and text-to-AMR parsing models accurately rely on sense IDs when producing either text or AMRs, respectively. Specifically, we examine how AMR-to-text generation models perform when the sense IDs are altered in the AMR graphs, and perform an analysis of the

[1] For example, `like-01` and `like-02` are two different senses of `like`, where `like-01` means *have affection towards, be fond of, enjoy (habitually)* while `like-02` means *would like, wish, want (polite)* (Palmer et al., 2005).

Proceedings of the 6th International Workshop on Designing Meaning Representations, pages 13–18
August 4, 2025. ©2025 Association for Computational Linguistics

accuracy of sense ID prediction in text-to-AMR parsers. We alter sense IDs in the input AMRs by:

- removing the sense IDs,
- replacing them with sense IDs that do not correspond with real PropBank frames,
- changing each sense ID to a realistic different sense of the same verb,
- swapping each verb's sense ID with the most (and least frequent) sense ID in the AMR3.0 corpus, and
- swapping each verb-particle construction with a verb (and the reverse: swapping all verbs with verb-particle constructions where possible).

We then generate text from four state-of-the-art AMR-to-text generation models on versions of the AMR3.0 dataset (Knight et al., 2020) with these sense ID alterations.

Next, we set out to ascertain the accuracy of sense IDs parsed by state-of-the-art automatic text-to-AMR parsers, which is related to the task of word sense disambiguation. We do this by examining whether the sense IDs match among the verbs that appear in both the automatically parsed AMRs and their human-annotated gold references.

We find that, while AMR-to-text generation models exhibit only a small decrease in automatic metric scores from these perturbations (removals and changes), there is still a statistically significant decrease for all models across all automatic metrics. We also find that, impressively, even for less frequently appearing senses, text-to-AMR parsers perform sense induction highly accurately. These results suggest that sense IDs are a contributing factor in the success of AMR technologies, but may be disregarded when necessary to promote multilingual extensions of AMR.

2 Methods

Here, we outline the data we use for experimentation (Section 2.1), the methods for sense ID alteration in AMR-to-text generation (Section 2.2), the evaluation techniques and models for AMR-to-text generation (Section 2.3), and the evaluation techniques and models for text-to-AMR parsing (Section 2.4).

2.1 Data

The AMR3.0 dataset contains 59,255 sentences written in English (from sources such as news and online forums), along with their matching gold (human-annotated) AMR graphs. We use only the test split of AMR3.0 to produce the altered datasets and generate parsed outputs, but identify the highest and lowest frequency sense IDs for each verb across the entire AMR3.0 dataset.

2.2 Sense ID Alterations

We evaluate the quality of AMR-to-text generation output under various conditions. We remove the sense IDs in four ways to observe how different components of a sense, such as the dash, signal the presence a predicate. We perform substitutions based on the frequency and existence of each individual sense ID to understand the effect of the appearance of senses in the training data. Lastly, we alter the verb-particle constructions to observe the impact of the verb form on the generated sentence.

Removed. We test removing sense IDs from AMR graphs in four ways: (1) completely remove the sense IDs and the dash preceding them (e.g. `get-01` to `get`), (2) remove the sense IDs but keep the dash preceding them (e.g. `get-01` to `get-`), (3) change all the sense IDs to `0` (e.g. `get-01` to `get-0`) , and (4) change all the sense IDs to 00 (e.g. `get-01` to `get-00`). We hypothesize that the dash functions as a marker for sense IDs, and therefore keeping the dash may improve sense induction performance compared to completely removing it, by signaling to the model that the preceding word is a predicate.

Arbitrarily large. We inspect the impact of a large sense ID that does not exist in PropBank by changing all the sense IDs to arbitrarily large numbers, randomized between 50 and 100, given that no sense IDs above 50 appear in PropBank.

Realistic substitution. Next, we change each sense ID to a random, "realistic" sense ID. If the word has multiple senses in PropBank, we substitute the current sense with another PropBank sense of the same verb form. If there is only one sense (`-01`), we substitute in `-02`.

Highest frequency. Here, we change each sense ID to the sense ID that appears most frequently for each verb in the AMR3.0 dataset. In the case of a tie (i.e. more than one sense has the same frequency), the lower numbered sense is used (given that it was added to PropBank first).

Lowest frequency. Similarly, we change each sense ID to the sense ID that appears the fewest number of times in the entire AMR3.0 dataset. In the case of a tie, the higher valued number is used

	amrlib			SPRING			BiBL			AMRBART		
Datasets	BERT	BLEU	MET.	BERT	BLEU	MET.	BERT	BLEU	MET.	BERT	BLEU	MET.
Baseline	0.9523	0.3869	0.7119	0.9589	0.4181	0.7366	0.9642	0.4753	0.7695	0.9651	0.4815	0.7732
Removed (1)	0.9512	0.3778	0.7074	0.9581	0.4126	0.7355	0.9631	0.4678	0.7674	0.9611	0.4399	0.7572
Removed (2)	0.9514	0.3815	0.7097	0.9577	0.4115	0.7328	0.9630	0.4669	0.7660	0.9614	0.4423	0.7575
Removed (3)	0.9516	0.3807	0.7089	0.9446	*0.3531*	*0.6852*	0.9604	0.4531	0.7534	0.9564	0.4111	*0.7385*
Removed (4)	0.9517	0.3789	0.7101	0.9583	0.4134	0.7351	0.9635	0.4713	0.7677	0.9612	0.4373	0.7579
Arbitrarily Large	0.9509	0.3753	0.7068	**0.9584**	0.4125	**0.7366**	0.9624	0.4644	0.7634	0.9602	0.4286	0.7531
Realistic Substitution	0.9519	0.3806	0.7104	0.9578	0.4088	0.7319	0.9624	0.4667	0.7624	0.9611	0.4390	0.7557
Highest Frequency	**0.9521**	**0.3852**	0.7111	0.9583	**0.4150**	0.7343	**0.9639**	0.4729	0.7676	**0.9645**	**0.4761**	**0.7693**
Lowest Frequency	0.9520	0.3836	**0.7111**	0.9582	0.4123	0.7338	0.9637	**0.4758**	**0.7690**	0.9637	0.4711	0.7683
To VPC	*0.9348*	*0.3088*	0.6763	*0.9429*	0.3566	0.7041	*0.9495*	*0.4122*	*0.7430*	*0.9499*	*0.4056*	0.7425
Remove VPC	0.9407	0.3175	*0.6601*	0.9497	0.3785	0.7102	0.9575	0.4451	0.7554	0.9584	0.4469	0.7601

Table 1: AMR-to-text generation results on the baseline and ten altered versions (VPC=verb-particle construction). The highest non-baseline scores within each model are bolded in blue, and the lowest scores for each model are italicized in red.

(given that it was added to PropBank later).

Change to verb-particle construction. Where possible, we change each verb to a verb-particle construction, such as `get-away-08` or `run-out-05`. To test the significance in changing a verb to a verb-particle construction, we exclude all AMR graphs that did not have any changes made (i.e.: verb has no verb-particle construction in PropBank). If there are valid senses to substitute, we choose one randomly. For example, if `drop-05` appears in the dataset, we replace it with a randomly chosen sense from the list: `drop-by-02`, `drop-off-03`, `drop-out-04`, `drop-in-08`. In this way, the parse is changed to have verb-particle construction (i.e. both the text and sense ID in the concept change) where applicable, though the verb-particle construction does not appear in the original sentence.

Remove verb-particle constructions. Finally, we change each verb-particle construction to a verb form, if applicable, using the same process as for changing to verb-particle constructions.

2.3 Generation Models & Evaluation

For AMR-to-text generation, we leverage four models: amrlib[2], SPRING (Bevilacqua et al., 2021), AMRBART (Bai et al., 2022), and BiBL (Cheng et al., 2022). For evaluation, we use the test set from AMR3.0 (Knight et al., 2020), which contains 1,898 AMR graphs, as some of these models were trained on the training portion of the corpus.

To analyze the effect of modifying sense IDs, we alter each node in the AMR graphs in the specified manner and then generate text from each of these sets of altered AMRs, using the aforementioned four generation models. We also generate baseline outputs from the original test split to compare how well our modified outputs perform.

We evaluate the generated text with BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and BERTscore (Zhang et al., 2020).

2.4 Parsing Models & Evaluation

For text-to-AMR parsing, we assess the accuracy of the sense IDs included in automatically produced AMR graphs. We use five models: the BART-large fine-tuned model of amrlib, SPRING, AMRBART, BiBL, and LeakDistill (Vasylenko et al., 2023). For evaluation, we use the test set from AMR3.0. Specifically, we use the Consensus dataset, which contains 100 AMRs, which we chose due to its suitable size for manual qualitative analysis.

In order to perform a small-scale analysis of text-to-AMR parser accuracy for sense IDs, we use the aforementioned parsers to generate the 100 predicted AMRs for each model. Then, we check for matching verbs, and of those verbs, correct sense IDs. For example, if the gold annotation is `get-01` and the predicted sense is `get-02`, then we have a matching verb and a different sense ID.

3 Results

Table 1 shows the results of our experiments on the effect of altering sense IDs on AMR-to-text generation; Table 2 contains the results of our evaluation of the sense ID accuracy of text-to-AMR parsing models.

3.1 Generation Results

We find that the automatic metric scores are only slightly—though consistently—lower for texts gen-

[2] amrlib GitHub Repository

erated from the altered datasets. This includes cases where sense IDs were removed, a promising finding for extending AMR-to-text generation to languages with insufficient PropBank frames.

Interestingly, the impact is more pronounced for better-performing models, suggesting they may be utilizing the sense ID information to a greater degree. In particular, AMRBART is the best-performing model with a baseline BERTscore of 0.9651, but its modified outputs show an average decrease of 0.0053. On the other hand, amrlib, SPRING, and BiBL have baseline BERTscores of 0.9523, 0.9589, and 0.9642, respectively, but their modified outputs show decreases of only 0.0035, 0.0045, and 0.0033 on average.

The text generated from the AMR nodes swapped with their highest frequency sense IDs has the highest automatic metric scores overall, with BERTscore decreases of just 0.0002 to 0.0006 compared to the baseline. This supports our hypothesis that AMR-to-text generation models tend to prioritize generating PropBank sense IDs based on their frequency in the AMR3.0 corpus. Notably, the AMRs swapped with the least frequent senses also perform competitively, occasionally outperforming all the other altered datasets in BLEU and METEOR scores. The highest and lowest frequency substitutions are the only alterations which ensure that all sense IDs present in the AMRs actually exist in PropBank, suggesting that maintaining valid sense information (and the same verb form) leads to higher quality text generation.

In contrast, AMRs involving verb-particle construction substitutions result in the greatest performance drops overall, with an average BERTscore decrease of 0.0122 across all models. These are the only cases where the root verbs change entirely, indicating that such changes disrupt the performance of AMR-to-text generation models more than changes to sense IDs alone.

We also find that the way in which the sense IDs are removed has an impact on the generated text, where maintaining the dash preceding the sense ID or changing the sense ID to `00` improves model performance compared to removing them both completely. This suggests that models treat the dash as a predicate marker. Furthermore, using `00` preserves the familiar formatting of most sense IDs and aligns with its use as a placeholder for missing predicates (Banarescu et al., 2019).

Though on an item-level basis the decrease in BERTscore is minimal, we find that *all* perturbations result in a statistically significant decrease in BERTscore when compared via t-tests ($p \leq 0.05$). We perform paired t-tests comparing the baseline BERTscore values against all datasets, except for the verb-particle construction changes, for which we perform unpaired t-tests given that these datasets are smaller (since not all individual AMR graphs were able to have a verb-particle construction substitution for any nodes).[3] This suggests that AMR-to-text generation models are still sensitive to changes in verb senses.

Models	Matching Verbs	Sense Accuracy (%)	1-Sense Verbs (%)
amrlib	351	98.0%	51.6%
SPRING	349	98.0%	51.6%
BiBL	341	98.5%	52.8%
AMRBART	350	98.9%	51.7%
LeakDistill	343	98.3%	52.5%

Table 2: Text-to-AMR parsing results. Sense Accuracy refers to instances where not only the root verbs but also the associated sense IDs are predicted correctly. About half of these matching verbs for each model have only one sense, with the exact percent for each model indicated here with the "1-Sense Verbs" column.

3.2 Parsing Qualitative Analysis

We check the sense accuracy of verbs which appear in both the gold AMR and the system output. As seen in Table 2, all five text-to-AMR parsers—amrlib, SPRING, BiBL, AMRBART, and LeakDistill—demonstrate high accuracy in assigning sense IDs to correctly predicted verbs, with accuracy rates from 98.0% to 98.9%. About half of these matching verbs for each model have only one sense, contributing to this high accuracy.

Impressively, the parsers also correctly identify less frequent senses. For instance, all five models accurately predict `run-04` in a sentence about Route 288 in Virginia,[4] even though `run-04` appears only 19 times in the AMR3.0 training split—compared to 188 instances of `run-01` and 149 of `run-02`. However, one of those 19 instances mentions "Virginia_State_Route_203" in a similar context, suggesting that the models drew on contextual patterns from training.

Our study is conducted using the AMR3.0 corpus, which primarily consists of newswire and on-

[3] For amrlib, the p-values range from <0.0001 to 0.0317. For SPRING, the p-values range from <0.0001 to 0.0244. For BiBL, the p-values range from <0.0001 to 0.0050. Finally, for AMRBART, the p-values range from <0.0001 to 0.0047.

[4] "Route 288, the circumferential highway running around the south - western quadrant of the Richmond New Urban Region, opened in late 2004."

line text, raising the question of how our findings on sense ID sensitivity generalize to other domains. From our results, we find that text-to-AMR parsers perform sense induction accurately even for senses that appear infrequently in the training data. This is promising for applying parsing models to other corpora, such as *The Little Prince* dataset (Banarescu et al., 2013), which is a literary work with often uncommon language usage. Even if infrequent senses appear in other corpora, our findings suggest that the parsing models would still perform well. The relatively small decrease in generation quality from sense ID alterations suggests that generation models are not effectively using the sense ID information. It is unclear whether this is due to the model architecture or how the sense ID information appears in AMR graphs. However, we know that the presence of a dash improves performance, suggesting that models recognize this as a signal to expect sense IDs. Additionally, the substantial drop in performance when substituting for verb-particle construction indicates that the verb form has a larger impact than the sense ID itself.

4 Conclusion & Future Work

In this work, we explored to what degree AMR-to-text generation models rely on sense IDs in AMR graphs, by swapping or removing the sense IDs in the nodes, and assessing the quality of the resulting text. We find that AMR-to-text generation models are susceptible to sense perturbations and suffer a small decrease in automatic metric scores (BERTscore, BLEU, and METEOR), with BERTscore decreases of up to 0.0175; though the decrease is relatively small, all of the changes that we make to the sense IDs result in a statistically significant decrease in text quality for all generation models. We also measured the accuracy of sense annotation in text-to-AMR parsers, and our parsing analysis reveals that AMR technologies do accurately perform sense induction when parsing.

Our results indicate that sense IDs enable higher quality text generation when included in the AMRs for AMR-to-text generation models, and provide insightful semantic content within the AMR. Still, the technical relevance of sense IDs is small, and may be worth avoiding if the creation of in-language frames precludes the development a non-English AMR extension—or for multilingual extensions of AMR broadly. Accordingly, our findings motivate future work investigating multilingual extensions of AMR that do not include any sense IDs and generalize roles across (i.e. moving from opaque arguments such as `:ARG0` to more generalizable terms such as `:agent`); finding generic terms that would be sufficiently representative across languages presents an additional challenge.

Acknowledgments

Thank you to our reviewers and members of the Amherst College ACNLP lab for feedback on our draft. This work is supported by the Amherst College HPC, which is funded by NSF Award 2117377.

References

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

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

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2019. Abstract Meaning Representation (AMR) 1.2.6 specification. https://github.com/amrisi/amr-guidelines/blob/master/amr.md.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. 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 AAAI*.

Ziming Cheng, Zuchao Li, and Hai Zhao. 2022. BiBL: AMR parsing and generation with bidirectional Bayesian learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5461–5475, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Lydia Feng, Gregor Williamson, Han He, and Jinho D. Choi. 2023. Widely interpretable semantic representation: Frameless meaning representation for broader applicability. *Preprint*, arXiv:2309.06460.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and 1 others. 2020. Abstract Meaning Representation (AMR) Annotation Release 3.0. Technical Report LDC2020T02, Linguistic Data Consortium, Philadelphia, PA.

Sebastian Padó. 2007. *Cross-lingual annotation projection models for role-semantic information*. Ph.D. thesis, Saarland University.

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

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

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

Lonneke van der Plas, Tanja Samardžić, and Paola Merlo. 2010. Cross-lingual validity of PropBank in the manual annotation of French. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 113–117, Uppsala, Sweden. Association for Computational Linguistics.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, and 1 others. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3):343–360.

Pavlo Vasylenko, Pere-Lluís Huguet Cabot, Abelardo Carlos Martínez Lorenzo, and Roberto Navigli. 2023. Incorporating graph information in transformer-based amr parsing. In *Findings of ACL*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *Proc. of ICLR*, Online.