Linguistic Guidance for Sequence-to-Sequence AMR Parsing

Binghao Tang, Boda Lin, Si Li*
School of Artificial Intelligence, Beijing University of Posts and Telecommunications
{tangbinghao,linboda,lisi}@bupt.edu.cn

Abstract

The Abstract Meaning Representation (AMR) parsing aims at capturing the meaning of a sentence in the form of an AMR graph. Sequence-to-sequence (seq2seq)-based methods, utilizing powerful Encoder-Decoder pre-trained language models (PLMs), have shown promising performance. Subsequent works have further improved the utilization of AMR graph information for seq2seq models. However, seq2seq models generate output sequence incrementally, and inaccurate subsequence at the beginning can negatively impact final outputs, also the interconnection between other linguistic representation formats and AMR remains an underexplored domain in existing research. To mitigate the issue of error propagation and to investigate the guiding influence of other representation formats on PLMs, we propose a novel approach of **L**inguistic **G**uidance for **S**eq2seq **A**MR parsing (**LGSA**). Our proposed LGSA incorporates the very limited information of various linguistic representation formats as guidance on the Encoder side, which can effectively enhance PLMs to their further potential, and boost AMR parsing. The results on proverbial benchmark AMR2.0 and AMR3.0 demonstrate the efficacy of LGSA, which can improve seq2seq AMR parsers without silver AMR data or alignment information. Moreover, we evaluate the generalization of LGSA by conducting experiments on out-of-domain datasets, and the results indicate that LGSA is even effective in such challenging scenarios.

1 Introduction

Abstract Meaning Representation (AMR) (Banarescu et al., 2013) is a semantic representation framework that systematically converts the meaning of a sentence into a rooted, labeled, and directed acyclic graph (DAG), where the nodes represent concepts and the edges denote semantic relations. Figure 1 depicts an example of an AMR graph. AMR has gained significant traction in various downstream natural language processing (NLP) tasks, including Information Extraction (Rao et al., 2017; Huang et al., 2016; Zhang and Ji, 2021), Text Summarization (Hardy and Vlachos, 2018; Liao et al., 2018), Machine Translation (Song et al., 2019), and Dialog System (Bonial et al., 2020; Bai et al., 2021).

The methods of AMR parsing can be divided into four categories: two-stage parsing (Flanigan et al., 2014; Flanigan et al., 2016; Lyu and Titov, 2018; Zhang et al., 2019a), transition-based parsing (Wang et al., 2015; Damonte et al., 2017; Vilares and Gómez-Rodríguez, 2018; Liu et al., 2018; Naseem et al., 2019; Fernandez Astudillo et al., 2020; Zhou et al., 2021a; Zhou et al., 2021b), graph-based parsing (Zhang et al., 2019b; Cai and Lam, 2019; Cai and Lam, 2020) and sequence-to-sequence (seq2seq)-based parsing (Barzdins and Gosko, 2016; Konstas et al., 2017; Peng et al., 2017; Ge et al., 2019; Bevilacqua et al., 2021; Yu and Gildea, 2022; Bai et al., 2022). In contrast to other approaches, seq2seq-based methods view AMR parsing as the process of transforming the input sentence into a linearized sequence of AMR graph. This allows for the seamless integration of powerful pre-trained seq2seq Transformer models, which have shown significant advancements in various NLP tasks. As a result,

*Corresponding author

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 1212-1222, Taiyuan, China, July 25 - 28, 2024.
Volume 1: Main Conference Papers
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

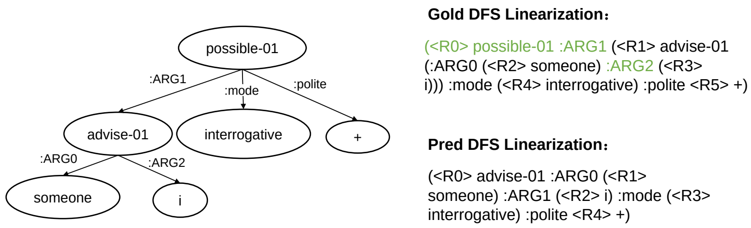

Figure 1: An AMR graph for sentence "Can someone give me advice please?" with its corresponding gold and predicted DFS-based linearized sequence. The prediction has the wrong root token and leads to a worse AMR graph compared with the gold AMR graph.

seq2seq-based methods have achieved remarkable performance, positioning them at the forefront of current research in AMR parsing.

Seq2seq-based works employ various techniques and auxiliary features while the performance is still unsatisfactory. For instance, structure-aware encoding (Ge et al., 2019) explicitly models syntax and semantics through complex source sequence linearization of the whole syntactic tree with semantic roles based on pure seq2seq Encoder-Decoder Transformer. More recently, seq2seq-based methods applying Pre-trained Language Models (PLMs) like BART (Lewis et al., 2020), further improve seq2seq-based methods by a large margin. These approaches focus on enhancing AMR parsers from various perspectives, such as AMR graph linearization (Bevilacqua et al., 2021), revised Decoder Attention (Yu and Gildea, 2022), curriculum learning (Wang et al., 2022), intermediate-task learning (Chen et al., 2022), graph pre-training (Bai et al., 2022), word-to-node alignment (Vasylenko et al., 2023) and instruction-tuning (Lee et al., 2023). What becomes apparent is that the methods before the adoption of encoder-decoder PLMs involved complex linearization and the utilization of the other whole linguistic representation formats. They also incorporated a significant amount of structural information that might not have been necessary for the AMR task, which could instead result in noise.

The subsequent approaches shifted their focus towards better harnessing the inherent structural information and alignments within the AMR data based on encoder-decoder pre-trained language models. However, the output AMR graph of seq2seq parsers is contingent upon the previously generated sub-graph, so the inaccurate sub-graph may lead to error propagation. Building upon the above issues, we aim to demonstrate that using the very limited information of different linguistic formats as guidance could still prompt the PLMs to generate better output tokens and reduce error propagation in AMR parsers, further improving their performance.

In this paper, we introduce a novel model-agnostic approach of **L**inguistic **G**uidance for **S**eq2seq **A**MR parsing (**LGSA**). The proposed LGSA leverages the very limited other linguistic representations as guidance on the encoder side and prompts the seq2seq-based parsers to generate more accurate sub-graphs. Specifically, we choose three commonly used linguistic representation formats: Part-of-Speech (POS) that can represent lexical features, Dependency Parsing (DP) that can represent syntactic features, and Semantic Dependency Parsing (SDP) (Oepen et al., 2015) that can represent semantic features. We build LGSA on the top of two recent seq2seq AMR parsers: SPRING (Bevilacqua et al., 2021) and Ancestors (Yu and Gildea, 2022). Experimental results on proverbial benchmark AMR2.0 and AMR3.0 demonstrate our proposed LGSA can further enhance the ability of the seq2seq AMR parsers. Also, Ancestors with LGSA of SDP guidance even achieves the comparable performance of AMRBART (Bai et al., 2022), which utilizes 200k silver AMR data for the post graph pre-training based on BART. We also conduct out-of-domain (OOD) experiments on three OOD datasets New3, TLP, and Bio, which confirm the effectiveness of the proposed LGSA method even in OOD situations.

2 Related Work

AMR Parsing AMR Parsing is a fundamental task that transforms a given sentence into a corresponding AMR graph. The methods of AMR parsing can be divided into four categories: two-stage parsing,

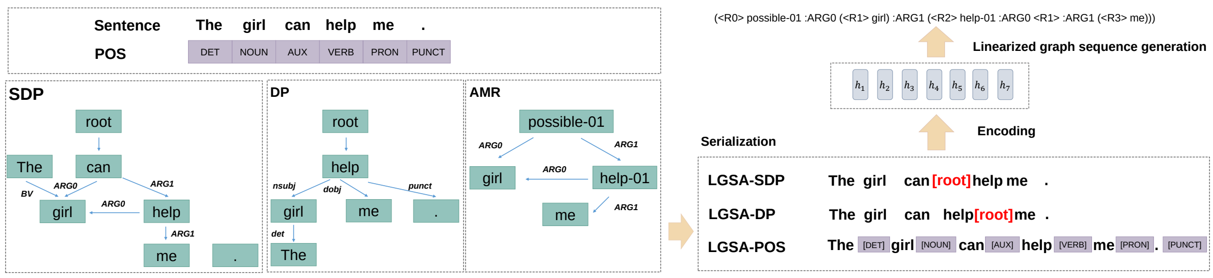

Figure 2: Serialization of our proposed LGSA, which utilizes the limited information of Semantic Dependency Parsing (SDP), Dependency Parsing (DP), and Part-of-Speech (POS) as the linguistic guidance to prompt PLMs. The obtained linguistic guidance source sequence is input to sequence-to-sequence AMR parsers to generate the corresponding linearized AMR graph.

transition-based parsing, graph-based parsing, and seq2seq-based parsing. Two-stage parsing first predicts potential concepts nodes, and then labels corresponding relations based on previous concepts decisions (Flanigan et al., 2014; Flanigan et al., 2016; Lyu and Titov, 2018; Zhang et al., 2019a). Transition-based methods (Wang et al., 2015; Fernandez Astudillo et al., 2020; Zhou et al., 2021a; Zhou et al., 2021b) process tokens of the input sentence from left to right and generate graph incrementally based on a pre-defined set of actions. Graph-based methods (Lyu and Titov, 2018; Zhang et al., 2019a; Cai and Lam, 2020) take AMR parsing as an incremental sequence-to-graph transduction task. Seq2seq-based methods treat AMR parsing as a sequence-to-sequence problem by AMR graph linearization. Several techniques have been employed in the early stages of many seq2seq works to improve the performance, such as auxiliary features (Konstas et al., 2017). For instance, (Ge et al., 2019) explicitly modeled syntax and semantics through linearized source sequence based on a syntactic tree with semantic roles and revised structure-aware self-attention. However, the performance of seq2seq-based methods has remained unsatisfactory until the introduction of SPRING (Bevilacqua et al., 2021). SPRING presented a better AMR graph linearization procedure and utilized BART (Lewis et al., 2020) as the backbone, which led to a significant improvement in seq2seq-based AMR Parsing. Subsequent seq2seq-based works have primarily focused on fully utilizing the structural information of AMR, such as adding ancestors information to Transformer Decoder (Yu and Gildea, 2022), utilizing different training methods like curriculum learning (Wang et al., 2022) and using extra silver data for intermediate-task learning (Chen et al., 2022) or further pretraining (Bai et al., 2022). In contrast, we pay attention to utilizing the very limited information of other linguistic formats to guide PLMs in a model-agnostic way.

Dependency Parsing and Semantic Dependency Parsing Dependency Parsing (DP) is a fundamental task in Natural Language Processing, aiming at representing the sentence in a format of tree structure. DP hopes to identify the syntactic relationships between words in a sentence. So DP is typically as syntactic features. The 2014 SemEval (Oepen et al., 2014; Oepen et al., 2015) shared task introduced a new representation of the sentence which focuses on the deep-semantic relations between words. Semantic dependency parsing involves converting an input sentence into a directed acyclic graph called a Semantic Dependency Graph (SDG). Due to the similarity between SDG and AMR graphs, which are both directed acyclic graphs, the structural information of SDG can serve as a supplement to AMR.

3 Methodology

3.1 Task Formulation

For seq2seq-based methods, the AMR parsing task is usually formulated as a transformation of the input sentence and linearized AMR sequence. Given input sentence $S = \{t_1, t_2, ..., t_n\}$, where t_i is the i-th token of the input sentence, model generates linearized AMR graph sequence $Y = \{y_1, y_2, ..., y_n\}$, where y_i is the i-th token of the AMR linearized sequence. As Figure 1 shows, the DFS-based linearization of SPRING (Bevilacqua et al., 2021) which uses special tokens like $\{\langle R_0\rangle, ..., \langle R_n\rangle\}$ to represent variables in the linearized graph, has been widely applied in seq2seq-based methods. However, in the case of the seq2seq model, the probability of generating the next token at the decoder is contingent upon the conditional probability of all previously generated tokens: $P(y_{i+1}|y_1, \ldots, y_i)$. Consequently, the accuracy of the initial token significantly influences the overall outcome. In this paper, we propose a novel model-agnostic approach of linguistic guidance for seq2seq AMR parsing, which provides the model with a modest amount of information on other linguistic representation formats as prompts to improve final AMR outcomes. We feed our linguistic serialized results into the Transformer Encoder-Decoder pre-trained language model instead of the original sentence. Then the model generates a corresponding AMR graph linearized sequence.

3.2 Linguistic Guidance Serialization

Serialization is a common technique used in seq2seq tasks, where various serializers have been designed for syntactic and semantic parsing (Ge et al., 2019; Lin et al., 2022). These serializers generate a serialized representation based on the specific linguistic representation, aiming at improving the performance of parsers.

Taking inspiration from these works, we hope to seek a straightforward approach to leverage a limited amount of other linguistic representation formats to guide seq2seq AMR parsers, which can be divided into parsing and lexical information[0]:

Parsing Information SDP and DP are selected as semantic and syntactic parsing information. SDP emphasizes semantics and converts the sentence into a DAG while DP focuses on dependency relations between words and represents the sentence in a format of tree structure. As shown in Figure 2, the input sentence "The girl can help me." has three forms of representation, SDP, DP, and AMR. The focal concept node of AMR form is *possible-01*, which corresponds to *can* in the original sentence. In the form of SDP, the root token of the sentence is *can* as well. This observation suggests that the root token of SDP has a natural alignment with the focal concept in AMR. Additionally, both SDP root tokens and AMR focal concepts are typically related to verbs, which serve a predicative function in sentences. Similarly, the syntactic root tokens of DP are also frequently linked to verbs like SDP and AMR. Thus we introduce a special token *[root]* to highlight the SDP and DP root tokens in the original sentences. We append this token to the respective root tokens and obtain LGSA-SDP and LGSA-DP sequences, as shown in Figure 2. This is done in the hope of choosing SDP and DP root tokens as soft guidance to prompt AMR parsers to generate more accurate tokens. Furthermore, based on the above procedures, we can simply serialize the source sequence using the root information of SDP and DP, without the need to employ a complete parsing tree for complex serialization or introduce numerous unnecessary parsing labels.

Lexical Information For lexical information, we select POS tagging, which is demonstrated beneficial for both semantic and syntactic parsing (Zhang et al., 2019a). To integrate the POS information into our serialization, we simply append the corresponding POS tag to the original token. Then the POS tag embedding can be added to the original sentence at the outset, as shown in Figure 2. To point out, compared to parsing information, we believe that lexical information may not be significantly helpful for AMR. Furthermore, POS information is not as easily utilized partially as parsing information. Therefore, we decided to fully utilize the entire POS information for comparison, to validate our aforementioned hypothesis.

In the aforementioned LGSA method, we can utilize a certain amount of external linguistic knowledge to preprocess the input sequence, highlighting the key information, and leveraging the powerful capabilities of PLMs to learn better AMR graph prediction.

[0]Different from (Ge et al., 2019), we aim to improve AMR parsers with limited utilization of extra-linguistic information based on PLMs.

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 1212-1222, Taiyuan, China, July 25 - 28, 2024.
Volume 1: Main Conference Papers

Datasets	**Train**	**Dev**	**Test**
AMR2.0	36,521	1,371	1,368
AMR3.0	55,635	1,722	1,898
New3	4,441	354	527
TLP	–	–	1,562
Bio	5,452	–	500

Table 1: Datasets statistics

4 Experiments

4.1 Datasets and Metrics

We evaluate our method on two proverbial AMR benchmarks, AMR2.0 (LDC2017T10) and AMR3.0 (LDC2020T02). Investigating OOD performance, we follow the same OOD experiment setting of SPRING (Bevilacqua et al., 2021), employ AMR2.0 training set, and evaluate on three OOD datasets, New3, Bio, and TLP. New3 is a subset of AMR3.0 but not contained in AMR2.0, which has 527 sentences. TLP is an AMR-annotated novel "The Little Prince" consisting of 1562 sentences. Bio is from the biomedical AMR corpus and has 500 sentences. The statistics are shown in Table 1.

We assess the performance of our parsers with Smatch (Cai and Knight, 2013) scores computed by amrlib [1] tools, which also provide fine-grained scores:

- Unlabeled: Smatch score after removing edge-labels
- NoWSD: Smatch score after ignoring Prop-bank senses (e.g., come-01 vs come-02)
- Concepts: F-score on the concept identification task
- Wikification (Wiki.): F-score on the wikification (:wiki roles)
- Named Entity Recognition (NER): F-score on the named entities (:name roles).
- Reentrancy (Reent.): Smatch score on reentrant edges.
- Negation (Neg.): F-score on the negation detection (:polarity roles)
- Semantic Role Labeling (SRL): Smatch score computed on :ARG-i roles only.

The fine-grained scores assess the predictions and gold AMR graphs from different angles.

4.2 Experiment Setting

For linguistic serialization, we choose Stanford NLP tools [2] to do tokenization and POS tagging. We use the existing parsing pipeline SuPar [3] to obtain the DM (DELPH-IN MRS) representation of the semantic dependency graph and dependency tree of the raw input sentence. We use BART-large (Lewis et al., 2020) as seq2seq pre-trained model backbone and RAdam (Liu et al., 2019) as optimizer. The learning rate is 3e-5, batch size is 1000, and beam size is 5. The epochs are 50 and the loss function is cross-entropy. The best model is selected according to performance on the validation set. All models are trained and evaluated on a single NVIDIA 3090Ti GPU. We conduct the same pre-processing and post-processing of the linearized AMR graph following SPRING (Bevilacqua et al., 2021). However, we do not adopt graph recategorization, which is usually used to alleviate data sparsity in graph-based methods yet has been proved to be harmful for OOD situations (Bevilacqua et al., 2021).

Model	Extra Data	Smatch	No WSD	Concept	NER	Neg.	Wiki.	Unlabeled	Reent.	SRL
AMR2.0										
HCL (Wang et al., 2022)	N	84.3	85.0	90.2	91.6	**75.9**	84.0	87.7	74.5	83.2
ATP(w/SRL) (Chen et al., 2022)	40k	85.2	85.6	90.7	**93.1**	74.9	**84.2**	88.3	**74.7**	**83.3**
AMRBART (Bai et al., 2022)	200k	**85.4**	**85.8**	**91.2**	91.5	74.0	81.4	**88.3**	73.5	81.5
SPRING (Bevilacqua et al., 2021)	N	83.8	84.4	90.2	90.6	74.4	**84.3**	86.1	70.8	79.6
SPRING(w/silver) (Bevilacqua et al., 2021)	200k	84.3	84.8	90.8	90.5	73.6	83.1	86.7	72.4	80.5
SPRING(w/LGSA-POS)	50k	84.2	84.8	90.7	90.8	74.6	83.9	86.4	71.9	80.1
SPRING(w/LGSA-DP)	50k	84.4	84.8	90.7	90.9	75.8	84.0	86.6	71.9	80.3
SPRING(w/LGSA-SDP)	50k	**84.5**	**85.0**	90.7	**91.3**	**76.2**	84.1	**86.8**	**72.4**	**80.5**
Ancestors (Yu and Gildea, 2022)	N	84.8	85.3	90.5	**91.8**	74.0	84.1	88.1	**75.1**	**83.4**
Ancestors(w/LGSA-POS)	50k	85.1	85.6	90.9	90.6	**75.1**	**84.7**	88.0	73.7	81.1
Ancestors(w/LGSA-DP)	50k	85.2	85.6	90.9	91.4	73.8	83.5	88.1	73.7	81.4
Ancestors(w/LGSA-SDP)	50k	**85.4**	**85.8**	**91.1**	91.3	74.6	83.9	**88.2**	74.1	81.5
AMR3.0										
HCL (Wang et al., 2022)	N	83.7	84.2	89.5	**89.0**	73.0	**82.6**	86.9	73.9	82.4
ATP(w/SRL) (Chen et al., 2022)	40k	83.9	84.3	89.7	88.4	**73.9**	81.0	87.0	**73.9**	**82.5**
AMRBART (Bai et al., 2022)	200k	**84.2**	**84.6**	**90.2**	88.5	72.1	78.9	**87.1**	72.4	80.3
SPRING (Bevilacqua et al., 2021)	N	83.0	83.5	89.8	87.2	**73.0**	**82.7**	85.4	70.4	78.9
SPRING(w/silver) (Bevilacqua et al., 2021)	200k	83.0	83.5	89.5	87.1	71.7	81.2	85.4	71.3	79.1
SPRING(w/LGSA-POS)	50k	83.5	83.8	89.8	87.1	72.4	81.9	85.9	71.0	78.7
SPRING(w/LGSA-DP)	50k	83.6	84.0	91.1	87.2	72.6	82.1	86.1	71.2	78.9
SPRING(w/LGSA-SDP)	50k	**83.8**	**84.1**	**91.3**	**88.0**	72.9	82.3	**86.3**	**71.4**	**79.1**
Ancestors (Yu and Gildea, 2022)	N	83.5	84.0	89.5	88.9	72.6	81.5	86.6	**74.2**	**82.2**
Ancestors(w/LGSA-POS)	50k	83.9	84.4	90.3	88.7	71.9	82.0	86.9	72.5	80.1
Ancestors(w/LGSA-DP)	50k	84.0	84.4	90.2	88.4	73.1	81.7	86.9	72.7	80.2
Ancestors(w/LGSA-SDP)	50k	**84.2**	**84.6**	**90.5**	**89.0**	**73.1**	**82.2**	**87.2**	72.7	80.4

Table 2: Experimental results on AMR2.0 and AMR3.0, our proposed LGSA does not fully leverage 50k extra data but rather only utilizes partial information as guidance to prompt PLMs.

4.3 Main results

We build our method on the top of seq2seq AMR parsers: SPRING (Bevilacqua et al., 2021) and Ancestors (Yu and Gildea, 2022), and show the improvement of our method. We also present the experimental results of recent seq2seq-based methods. To point out, the compared methods ATP (Chen et al., 2022) with Semantic Role Labeling (SRL) uses 40k SRL data from OntoNotes (LDC2013T19) for intermediate-task learning and AMRBART (Bai et al., 2022) leverage 200k silver data from Gigaword (LDC2011T07) parsed by SPRING for further graph pretraining on the basis of vanilla BART while our proposed LGSA method obtain extra-linguistic information only based on the original sentences and do not require any additional corpora.

The main results are as shown in Table 2. Among the three types of linguistic information, LGSA-SDP shows the highest improvement and LGSA-POS performs the worst. More precisely, SPRING with LGSA-SDP achieves an improvement of 0.7 Smatch on AMR2.0 and 0.8 Smatch on AMR3.0, and Ancestors with LGSA-SDP achieves an improvement of 0.6 Smatch on AMR2.0 and 0.7 Smatch on AMR3.0 compared with baseline models. The results suggest that incorporating semantic parsing information (SDP) is more helpful for AMR parsing than incorporating lexical information (POS). This is likely because AMR is a semantic representation, therefore, semantic features should be more relevant to capturing its meaning. Furthermore, the results show that Ancestors with LGSA-SDP achieves competitive performance even against models that utilize a large amount of silver AMR data such as AMRBART (Bai et al., 2022) and ATP (Chen et al., 2022). This indicates that our approach can achieve promising results without requiring any additional corpora, but simply by leveraging the linguistic features extracted from the original sentences and utilize a limited amount of information to be guidance on the encoder side. For fine-grained scores, Ancestors with LGSA-SDP also obtain the best result of No

[1] https://github.com/bjascob/amrlib
[2] https://github.com/stanfordnlp/stanza
[3] https://github.com/yzhangcs/parser

Model	New3	TLP	Bio
SPRING (Bevilacqua et al., 2021)	73.7	77.3	59.7
SPRING(w/LGSA-POS)	73.9	77.5	61.0
SPRING(w/LGSA-DP)	74.0	77.5	61.2
SPRING(w/LGSA-SDP)	**74.1**	**77.7**	**61.4**
Ancestors (Yu and Gildea, 2022)	75.2	79.3	61.4
Ancestors(w/LGSA-POS)	75.3	79.3	61.6
Ancestors(w/LGSA-DP)	75.4	79.4	61.8
Ancestors(w/LGSA-SDP)	**75.6**	**79.4**	**62.0**

Table 3: Experimental results on OOD datasets.

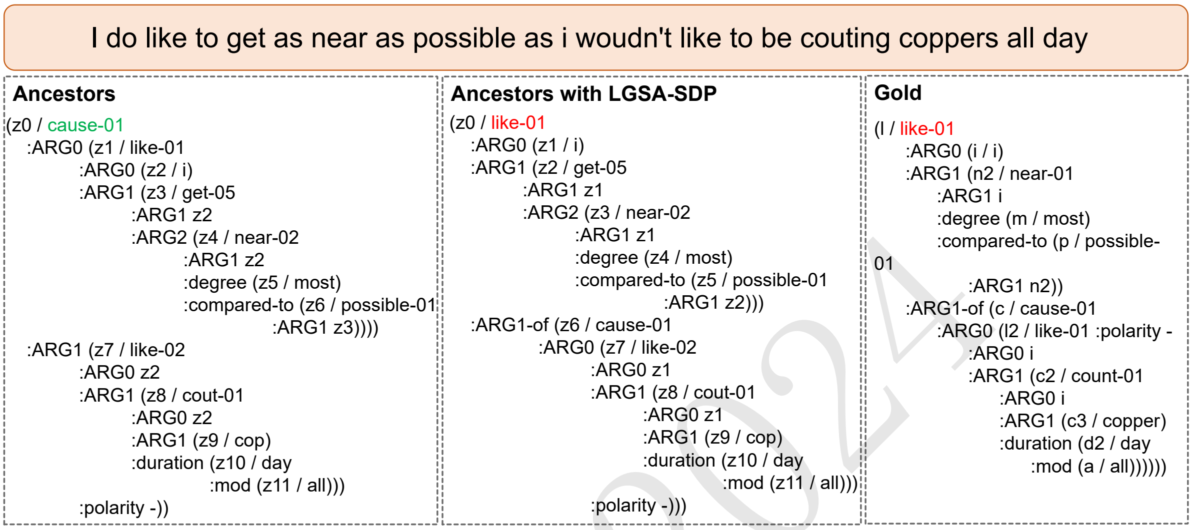

Figure 3: A case selected from AMR2.0 test set. For the input sentence, the baseline model Ancestors with LGSA-SDP achieves the right root token and a more accurate AMR graph, while Ancestors (Yu and Gildea, 2022) gets a relatively worse result. The results show that using Ancestors alone yields a Smatch score of 63.2 while incorporating LGSA-SDP improves it to 73.7.

WSD on AMR2.0 and No WSD, Concept, NER and Unlabeled on AMR3.0.

The main experimental results prove that our proposed LGSA method can enhance the performance of the current seq2seq AMR parsers by utilizing a limited amount of external linguistic knowledge, which aligns with our initial intention stated in the introduction. It also demonstrates that linguistic representations are an interconnected whole, where different forms of representation mutually benefit each other. The closer the forms of representation, the more pronounced this beneficial effect, as exemplified by SDP and AMR.

4.4 OOD Results

Previous methods have demonstrated promising results on in-domain datasets, but their performance significantly declines on OOD datasets, highlighting the importance of generalization in such methods. In this paper, we compare our proposed LGSA method with our baseline seq2seq parsers. As illustrated in Table 3, LGSA-SDP still achieves the best results among all the methods on three OOD datasets whether it is SPRING or Ancestors. Also, LGSA-DP performs worse and LGSA-POS performs the worst, which is consistent with their performance on the main in-domain dataset. These results demonstrate the efficacy of our proposed LGSA method in utilizing other linguistic representation formats to guide AMR parsers and enhance performance in OOD scenarios. Yet we believe that the limited performance improvement on OOD datasets is due to the accuracy of the parsing pipeline on OOD data. Given the use of very limited information, the accuracy like root tokens limits the performance of AMR parsers in

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 1212-1222, Taiyuan, China, July 25 - 28, 2024.
Volume 1: Main Conference Papers

OOD scenarios. Despite these challenges, our proposed LGSA method has still proven effective in OOD scenarios across multiple runs of experiments.

4.5 Case Study

We further conduct a case study on the seq2seq parser Ancestors (Yu and Gildea, 2022) to explore the output results with and without incorporating our method, thereby proving the effectiveness of our approach. Given that the LGSA-SDP shows the best improvement on both main and OOD results, we compare the output AMR graphs of the baseline model Ancestors (Yu and Gildea, 2022) and Ancestors (Yu and Gildea, 2022) with LGSA-SDP.

As is shown in Figure 3, Ancestors (Yu and Gildea, 2022) with LGSA-SDP gets the right focus concept *like-01* with the aid of SDP, and generates a more accurate AMR graph while Ancestors gets the wrong focus concept *cause-01* and results in less correct structure. The baseline prediction gets Smatch 63.2 computed with gold graph while ours obtains Smatch 73.7. This indicates that with SDP as an additional guidance, compared to the original output, the seq2seq AMR parser Ancestors can better predict the correct root token. This leads to a more accurate prediction of the corresponding AMR graph, thereby enhancing the performance of AMR parsers. In this particular sample case, with only the very limited additional guidance provided like the SDP root token, the model shows a ten-point improvement in the Smatch score. This also demonstrates that for AMR parsers based on PLMs, the use of limited external linguistic knowledge can significantly enhance their current performance.

5 Conclusions

In this paper, we focus on the error propagation of seq2seq parsers and investigate to utilize a limited amount information of from other linguistic representation formats as guidance for seq2seq AMR parsing to better utilize pre-trained language models. We propose a novel model-agnostic approach called **L**inguistic **G**uidance for **S**eq2seq **AMR** Parsing (**LGSA**), which leverages Part-of-Speech (POS) information for lexical features, Dependency Parsing for syntactic features, and Semantic Dependency Parsing for semantic features. Experiments on AMR2.0 and AMR3.0 demonstrate that LGSA can incorporate additional linguistic information on the Encoder side and enhance the ability of AMR parsers. Furthermore, we also demonstrate the effectiveness of LGSA in out-of-domain (OOD) benchmarks including New3, TLP, and Bio.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (NSFC) No. U23B2052, Program for Youth Innovative Research Team of BUPT No. 2023QNTD02.

References

Xuefeng Bai, Yulong Chen, Linfeng Song, and Yue Zhang. 2021. Semantic representation for dialogue modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4430–4445, Online, August. Association for Computational Linguistics.

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, May. 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, August. Association for Computational Linguistics.

Guntis Barzdins and Didzis Gosko. 2016. RIGA at SemEval-2016 task 8: Impact of Smatch extensions and character-level neural translation on AMR parsing accuracy. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1143–1147, San Diego, California, June. 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.

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, May. European Language Resources Association.

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, August. Association for Computational Linguistics.

Deng Cai and Wai Lam. 2019. Core semantic first: A top-down approach for AMR parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3799–3809, Hong Kong, China, November. Association for Computational Linguistics.

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, July. Association for Computational Linguistics.

Liang Chen, Peiyi Wang, Runxin Xu, Tianyu Liu, Zhifang Sui, and Baobao Chang. 2022. ATP: AMRize then parse! enhancing AMR parsing with PseudoAMRs. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2482–2496, Seattle, United States, July. 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, April. Association for Computational Linguistics.

Ramón Fernandez Astudillo, Miguel Ballesteros, Tahira Naseem, Austin Blodgett, and Radu Florian. 2020. Transition-based parsing with stack-transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1001–1007, Online, November. Association for Computational Linguistics.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland, June. Association for Computational Linguistics.

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

DongLai Ge, Junhui Li, Muhua Zhu, and Shoushan Li. 2019. Modeling source syntax and semantics for neural amr parsing. In *IJCAI*, pages 4975–4981.

Hardy Hardy and Andreas Vlachos. 2018. Guided neural language generation for abstractive summarization using Abstract Meaning Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 768–773, Brussels, Belgium, October-November. Association for Computational Linguistics.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and Avirup Sil. 2016. Liberal event extraction and event schema induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 258–268, Berlin, Germany, August. Association for Computational Linguistics.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada, July. Association for Computational Linguistics.

Young-Suk Lee, Ramón Fernandez Astudillo, Radu Florian, Tahira Naseem, and Salim Roukos. 2023. Amr parsing with instruction fine-tuned pre-trained language models. *arXiv preprint arXiv:2304.12272*.

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, July. Association for Computational Linguistics.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract Meaning Representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Boda Lin, Zijun Yao, Jiaxin Shi, Shulin Cao, Binghao Tang, Si Li, Yong Luo, Juanzi Li, and Lei Hou. 2022. Dependency parsing via sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7339–7353, Abu Dhabi, United Arab Emirates, December. Association for Computational Linguistics.

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

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*.

Chunchuan Lyu and Ivan Titov. 2018. AMR parsing as graph prediction with latent alignment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 397–407, Melbourne, Australia, July. Association for Computational Linguistics.

Tahira Naseem, Abhishek Shah, Hui Wan, Radu Florian, Salim Roukos, and Miguel Ballesteros. 2019. Rewarding Smatch: Transition-based AMR parsing with reinforcement learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4586–4592, Florence, Italy, July. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72, Dublin, Ireland, August. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, and Zdeňka Urešová. 2015. SemEval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926, Denver, Colorado, June. Association for Computational Linguistics.

Xiaochang Peng, Chuan Wang, Daniel Gildea, and Nianwen Xue. 2017. Addressing the data sparsity issue in neural AMR parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 366–375, Valencia, Spain, April. Association for Computational Linguistics.

Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. 2017. Biomedical event extraction using Abstract Meaning Representation. In *BioNLP 2017*, pages 126–135, Vancouver, Canada,, August. Association for Computational Linguistics.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using amr. *Transactions of the Association for Computational Linguistics*, 7:19–31.

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 the Association for Computational Linguistics: ACL 2023*, pages 1995–2011, Toronto, Canada, July. Association for Computational Linguistics.

David Vilares and Carlos Gómez-Rodríguez. 2018. A transition-based algorithm for unrestricted AMR parsing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 142–149, New Orleans, Louisiana, June. Association for Computational Linguistics.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015. A transition-based algorithm for AMR parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375, Denver, Colorado, May–June. Association for Computational Linguistics.

Peiyi Wang, Liang Chen, Tianyu Liu, Damai Dai, Yunbo Cao, Baobao Chang, and Zhifang Sui. 2022. Hierarchical curriculum learning for AMR parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 333–339, Dublin, Ireland, May. Association for Computational Linguistics.

Chen Yu and Daniel Gildea. 2022. Sequence-to-sequence AMR parsing with ancestor information. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 571–577, Dublin, Ireland, May. Association for Computational Linguistics.

Zixuan Zhang and Heng Ji. 2021. Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–49, Online, June. Association for Computational Linguistics.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019a. AMR parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94, Florence, Italy, July. Association for Computational Linguistics.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019b. Broad-coverage semantic parsing as transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3786–3798, Hong Kong, China, November. Association for Computational Linguistics.

Jiawei Zhou, Tahira Naseem, Ramón Fernandez Astudillo, and Radu Florian. 2021a. AMR parsing with action-pointer transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5585–5598, Online, June. Association for Computational Linguistics.

Jiawei Zhou, Tahira Naseem, Ramón Fernandez Astudillo, Young-Suk Lee, Radu Florian, and Salim Roukos. 2021b. Structure-aware fine-tuning of sequence-to-sequence transformers for transition-based AMR parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6290, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 1212-1222, Taiyuan, China, July 25 - 28, 2024.
Volume 1: Main Conference Papers