Cross-domain Generalization for AMR Parsing

Xuefeng Bai[*♠], **Sen Yang**[♡], **Leyang Cui**[♣], **Linfeng Song**[♢], **Yue Zhang**[♠†]
[♠] School of Engineering, Westlake University, China
[♡] The Chinese University of Hong Kong, China
[♣] Tencent AI Lab, Shenzhen, China
[♢] Tencent AI Lab, Bellevue, WA, USA
[†] Institute of Advanced Technology, Westlake Institute for Advanced Study, China

Abstract

Abstract Meaning Representation (AMR) parsing aims to predict an AMR graph from textual input. Recently, there has been notable growth in AMR parsing performance. However, most existing work focuses on improving the performance in the specific domain, ignoring the potential domain dependence of AMR parsing systems. To address this, we extensively evaluate five representative AMR parsers on five domains and analyze challenges to cross-domain AMR parsing. We observe that challenges to cross-domain AMR parsing mainly arise from the distribution shift of words and AMR concepts. Based on our observation, we investigate two approaches to reduce the domain distribution divergence of text and AMR features, respectively. Experimental results on two out-of-domain test sets show the superiority of our method.

1 Introduction

Abstract meaning representation (AMR; Banarescu et al. 2013) is a broad-coverage semantic structure formalism that represents the meaning of a text in a rooted directed graph. As shown in Figure 1, the nodes in an AMR graph represent concepts such as entities and predicates, and the edges indicate their semantic relations. AMR parsing (Flanigan et al., 2014; Konstas et al., 2017; Lyu and Titov, 2018; Guo and Lu, 2018; Zhang et al., 2019a; Cai and Lam, 2020a; Bevilacqua et al., 2021; Zhou et al., 2021b; Bai et al., 2022a) is the task of transforming natural language into AMR graphs. This is a fundamental task in semantics, which can also benefit downstream use.

AMR has been proven to be useful for many downstream tasks, such as information extraction (Huang et al., 2016; Martínez-Rodríguez et al., 2020; Zhang and Ji, 2021; Luo et al., 2022; Chen et al., 2022b; Wang et al., 2022), text summarization (Liu et al., 2015; Liao et al., 2018; Chen et al., 2021, 2022c; He et al., 2022), machine translation (Song et al., 2019; Slobodkin et al., 2022; Chen et al., 2022a), text generation (Konstas et al., 2017; Song et al., 2018; Zhu et al., 2019; Bai et al., 2020; Ribeiro et al., 2021), and dialogue systems (Bonial et al., 2020; Bai et al., 2021, 2022b). To benefit such a diverse set of tasks that covers various domains, an ideal AMR parser should generalize well across different domains. However, most existing work only focuses on improving the in-domain parsing accuracy, ignoring the performances on other domains. Though state-of-the-art AMR parsers can obtain a SMATCH score of over 84% on an in-domain test set, we observe that their cross-domain performance is still weak (e.g., lower than 65% on the biomedical domain). It remains an open-question how well different types of AMR parsers generalize to out-of-domain (OOD) data.

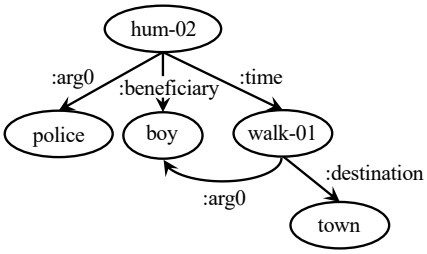


Figure 1: An AMR graph for sentence "*The police hummed to the boy as he walked to town.*"

In this work, we take the first step to study the cross-domain generalization ability of a range of typical AMR parsers, investigating three main research questions: 1) *how well do different AMR parsers perform on out-of-domain test sets?* 2) *what are the main challenges to cross-domain AMR parsing?* and 3) *how to improve the performance of cross-domain AMR parsing?*

We empirically choose five major AMR parsers for comparison, including a two-stage statistical

[*] Work done as an intern at Tencent AI Lab Seattle.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 10907 - 10921
December 7-11, 2022

parser (Flanigan et al., 2014), a graph-based parser (Cai and Lam, 2020b), a transition-based parser (Zhou et al., 2021b), a Seq2Seq-based parser (Bevilacqua et al., 2021), and an AMR-specific pre-training parser (Bai et al., 2022a). The test domains cover news, biomedical, novel, and wiki questions. We conduct experiments under the zero-shot setting, where a model is trained on the source domain and evaluated on the target domain without using any target-domain labeled data. Our results show that 1) all models give relatively lower (up to 45.5%) performances on out-of-domain test sets, with the most dramatic drop on named entities and wiki links; 2) the graph pretraining-based parser is stronger in domain transfer than the other parsers; 3) the transition-based parser is more robust than the seq2seq-based parser. We further analyze the impact of a set of linguistic features, and the results suggest that the performance degradation is positively correlated with the distribution shifts of words and AMR concepts. Compared with the distribution divergences of the input features, those of the output features are more challenging to cross-domain AMR parsing.

Based on our analysis, we investigate two approaches to bridge the domain gap for improving cross-domain AMR parsing. We first continually pre-train a BART model on target domain raw text to reduce the distribution gap of words. To further bridge the domain gap of output features, we adopt a pre-trained AMR parser to construct silver AMR graphs on the target domain, which potentially reduces the output features divergence. Experimental results show that the proposed methods consistently improve the parsing performance on out-of-domain test sets. To our knowledge, this is the first systematic study on cross-domain AMR parsing. Our code and results will be available at https://github.com/goodbai-nlp/AMR-DomainAdaptation.

2 Related Work

2.1 AMR Parsing

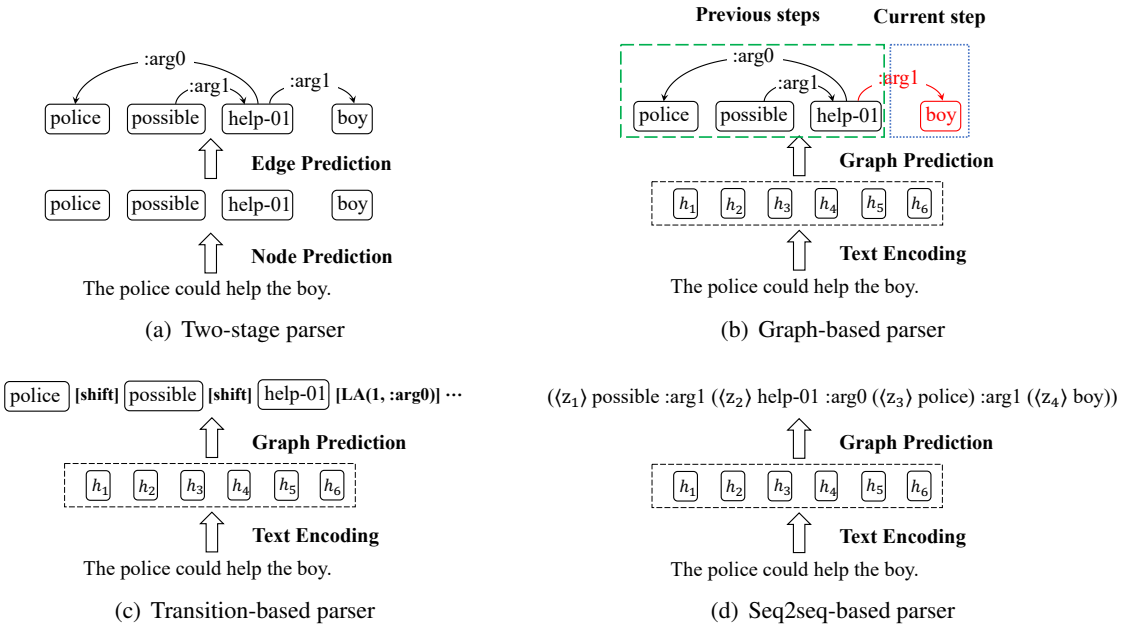
On a coarse-grained level, the current AMR parsing systems can be categorized into two main classes. The first is two-stage parsing system, which first identifies concepts, and then predicts relations based on the concept decisions. Two tasks are modeled either in a pipeline (Flanigan et al., 2014, 2016) or jointly (Lyu and Titov, 2018; Zhang et al., 2019a). The other one is one-stage parsing, which generates a parse graph incrementally. The one-stage parsing methods can be further divided into three categories: graph-based parsing, transition-based parsing, and seq2seq-based parsing. Transition-based parsing induces an AMR graph by predicting a sequence of transition actions. The transition-based AMR parsers either maintain a stack and a buffer (Wang et al., 2015; Damonte et al., 2017; Ballesteros and Al-Onaizan, 2017; Vilares and Gómez-Rodríguez, 2018; Liu et al., 2018; Naseem et al., 2019; Fernandez Astudillo et al., 2020; Lee et al., 2020) or make use of a pointer (Zhou et al., 2021a,b). Graph-based parsing builds a semantic graph incrementally. At each time step, a new node along with its connections to existing nodes are jointly decided. The graph is induced either in top-down manner (Cai and Lam, 2019) or in specific traversal order (Zhang et al., 2019b; Cai and Lam, 2020a). Seq2seq-based parsing treats AMR parsing as a sequence-to-sequence problem by linearizing AMR graphs so that existing seq2seq models can be readily utilized. Various seq2seq architectures have been employed for AMR parsing, such as vanilla seq2seq (Barzdins and Gosko, 2016; Konstas et al., 2017), supervised attention (Peng et al., 2017), character-based (Van Noord and Bos, 2017), and pre-trained Transformer (Bevilacqua et al., 2021; Bai et al., 2022a).

Despite great success, most previous work on AMR parsing focuses on the in-domain setting, where the training and test data share the same domain. In contrast, we systematically evaluate the model performance on 4 out-of-domain datasets. To our knowledge, we are the first to systematically study cross-domain generalization for AMR parsing.

2.2 Related Tasks

We summarize recent research studying other semantic formalisms as well as the cross-domain generalization of named entity recognition (NER), semantic role labeling (SRL) and constituency parsing.

Semantic parsing on other formalisms. AMR is strong-correlated with other semantic formalisms such as semantic dependency parsing (SDP, Oepen et al., 2016) and universal conceptual cognitive annotation (UCCA, Abend and Rappoport, 2013; Hershcovich et al., 2017), and recent



(a) Two-stage parser (b) Graph-based parser

(c) Transition-based parser (d) Seq2seq-based parser

Figure 2: Illustration of four AMR parsers given the input "*The police could help the boy.*".

researches show that they can be represented in a unified format and parsed by a generalized framework (Hershcovich et al., 2018; Zhang et al., 2019b). However, most of previous work focus on specific domain, leaving the study of cross-domain generalization unexplored.

Cross-domain NER. Named entity recognition (NER) is a subtask of AMR parsing. To build a robust NER system across domains, Yang et al. (2017) directly train NER models on the domain-mixed corpus. Wang et al. (2020) introduce an auxiliary task to predict the domain label. Recently, many studies focus on recognizing the unseen entity types in the target domain. Wiseman and Stratos (2019) and Yang and Katiyar (2020) propose distance-based methods, which copy the entity label of nearest neighbors. Cui et al. (2021) and Ma et al. (2021) adopt prompt-based methods by using BART and BERT, respectively.

Cross-domain SRL. SRL can also be seen as semantic-related subtasks of AMR parsing. Dahlmeier and Ng (2010) conduct an extensive study by analyzing various features and techniques that are used for SRL domain adaptation. Lim et al. (2014) combine a prior model with a structural learning model to build a multi-domain SRL system. Do et al. (2015) exploit the knowledge from a neural language model and external linguistic resource for domain adaptation on biomedical data. Rajagopal et al. (2019) develop a label mapping strategy and a layer adapting approach for cross-domain SRL. Compared with cross-domain NER and SRL, the task of cross-domain AMR parsing is more challenging since AMR is a graph formalism, and AMR contains more types of concepts and relations.

Cross-domain constituency parsing. Yang et al. (2022) investigated challenges to open-domain syntactic parsing, introducing datasets on new domains and analyzing the key factors on to cross-domain constituency parsing using a set of linguistic features. Our work is similar to their work in studying the key challenges on various parsing systems. However, we focus on AMR and conducts fine-grained semantic-related evaluation. In addition, we provide a intuitive solution for improving cross-domain AMR parsing.

3 Compared Models

We choose the representative or top-performing parser of two-stage, graph-based, transition-based, seq2seq-based as well as a pre-trained parser for evaluation. In particular, the following AMR parsing systems are considered:

JAMR (Flanigan et al., 2014), as shown in Figure 2(a), is a two-stage parsing model which predicts concepts and relations in a pipeline. JAMR identifies concepts and predicts the relations using two discriminatively-trained linear structured predictors, which use rich features

Models	Categratory	Pre-proc.	Post-proc.	Ext. Data	PLM
JAMR	Two-stage	✓	✓	POS, train align, etc.	✗
AMRGS	Graph	Recat.	concept, polarity, wiki	POS, NER, Lemm.	BERT
StructBART	Transition	✗	wiki	train align.	BART
SPRING	Seq2seq	✗	wiki	✗	BART
AMRBART	Pretrain + Seq2seq	✗	wiki	200k silver	BART

Table 1: Compared AMR parsing systems. "Recat"–graph re-categorization.

like part-of-speech tagging (POS), named entities recognition (NER), lemmatization, etc. In addition, JAMR relies on an external aligner to construct supervision signals for both stages.

AMRGS (Cai and Lam, 2020a) is a graph-based parser which builds a semantic graph incrementally. As shown in Figure 2(b), at every step, the graph-based parser predicts one node and its connection to existing graph. AMRGS learns mutual causalities between text and graph by updating the sentence and graph representations iteratively. AMRGS obtains word-level representation from a pre-trained language model (i.e., BERT (Devlin et al., 2019)) and uses POS, NER and lemmatization as external knowledge to make predictions.

StructBART (Zhou et al., 2021b), as shown in Figure 2(c), is a transition-based parser which generates an AMR graph through a sequence of transition actions. In particular, the transition actions are:

- **SHIFT** moves token cursor to right.
- **<string>** creates a node of name <string>.
- **COPY** creates a node with the name of the cursor-pointed token.
- **LA(j, LBL)** creates an *arc* with label LBL from the last generated node to the jth generated node.
- **RA(j, LBL)** is same as LA but with reversed edge direction.
- **ROOT** assigns the last generated node as root.

StructBART takes a pre-trained BART model as the backbone and extends the original vocabulary with transition actions. Additionally, StructBART requires an external aligner to obtain oracle transition actions for training.

SPRING (Bevilacqua et al., 2021), as shown in Figure 2(d), is a sequence-to-sequence parser which transforms a text sequence into a linearized AMR sequence. SPRING adopts a depth-first algorithm to transform AMR graphs into a sequence where concepts and relations are treated equally. To deal with co-referring nodes, SPRING adds special tokens to the vocabulary. Same with StructBART, SPRING also initializes model parameters with BART.

AMRBART (Bai et al., 2022a) is a continually pre-trained BART model on AMR graphs and text. It uses three graph-based pre-training tasks to improve the structure awareness of the encoder and decoder and another four tasks that jointly learns on text and AMR graph to capture the correspondence between AMR and text. AMRBART is pre-trained on 250k training instances, which lie in the same domain as AMR2.0.

In addition, JAMR uses complicated rule-based pre-processing and post-processing steps to simplify the input and reconstruct the AMR graphs. AMRGS uses rule-based graph re-categorization for pre-processing and recovers concept sense tags, wiki links, and polarities during post-processing. StructBART, SPRING, and AMRBART do not require pre-processing steps and use the BLINK Entity Linker (Wu et al., 2020) to handle wiki links during post-processing. Table 1 summarizes the above systems according to their characteristics.

4 Experiments

Experimental configurations and our adopted datasets are shown in Sections 4.1 and 4.2, respectively. To study the cross-domain generalization ability of current AMR parsers, we first quantify the difference between in-domain training data and out-of-domain test data (Section 4.3), and then evaluate the cross-domain performance of 5 typical AMR parsers (Section 4.4).

4.1 Experimental Settings

Model Configuration. We adopt the officially released code of each system and use their default configuration to re-train and evaluate the model performance. The best model is selected according to the performance on the in-domain validation set. All models are trained and evaluated on a single Nvidia Tesla V100 GPU.

Metrics. We assess the performance of parsing models with Smatch (Cai and Knight, 2013)

Dataset	Category	Sents	Tokens
ID AMRs			
AMR2.0	train	36,521	653K
	dev	1,371	30K
	test	1,368	29K
OOD AMRs			
New3	train	4,441	83K
	dev	354	64K
	test	527	8K
TLP	test	1,562	21K
Bio	train	5,452	138K
	test	500	13K
QALD-9	test	558	5K
External Data			
Raw text (TLP)	-	109k	2M
Raw text (Bio)	-	200k	4.4M
Silver AMR (TLP)	-	109k	2M
Silver AMR (Bio)	-	200k	4.4M

Table 2: Dataset statistics.

scores computed with the *amrlib*[1] tools, which also report fine-grained scores including unlabeled, NoWSD, concept identification, NER, negations, reentrancy and wiki links.[2]

4.2 Datasets

In-Domain Dataset. We train and evaluate AMR parsers on standard benchmarks, which we refer to as the In-Domain (ID) setting. We use **AMR2.0** (LDC2017T10)[3] as ID dataset which consists AMRs from newswire, discussion forum and other web logs, web collections.

Out-of-Domain Datasets. We consider the following datasets for out-of-domain (OOD) evaluation: **New3**, a subset of AMR3.0[4], whose original source was the DARPA LORELEI program (Christianson et al., 2018). The domain of New3 is close to AMR2.0; **TLP**[5] is an annotation of the novel *The Little Prince* that contains 1,562 sentences. **Bio**[6], which consists of annotations of biomedical texts, including PubMed articles and sentences from other biological corpus. **QALD-9**[7] (Lee et al., 2022), a recent released dataset whose original source the questions of SQuAD2.0 (Rajpurkar et al., 2016). Since QALD-9 comprises only 150 test sentences, we concatenate the train and test set for evaluation, leading to 558 instances in total.

In addition, we collect raw text from two domains: biomedical data (like Bio) and fairy tales data (like TLP). The former is sampled from PubMedQA (Jin et al., 2019) dataset, while the latter is a collection of fairy tales between the 19th century and early 20th. We also construct silver data for TLP and Bio by employing a state-of-the-art AMR parser (Bai et al., 2022a) to parse collected raw sentences into AMR graphs. Table 2 shows more details of above datasets.

4.3 Distributional Variance Across Datasets

To better understand the cross-domain parsing performance, we quantify the difference between the ID training set and 5 test sets according to the following list of linguistic features: **input text features**, including input length, uni-gram, bi-gram, tri-gram; **output AMR features**, which consists of AMR concepts, AMR relations, and $\langle \text{concept}, \text{relation}, \text{concept} \rangle$ triplets. We report the average score for input length. For other features, we follow Yang et al. (2022) to consider both the out-of-vocabulary (OOV) rate and the Jensen-Shannon divergence (Fuglede and Topsoe, 2004) to measure the difference. The former calculates the vocabulary difference between two domains, while the latter records the distributional divergence. Given a specific feature, denoting the feature distribution in the source domain as P and the distribution in the target domain as Q, the Jensen-Shannon divergence (*JS*) is calculated as:

$$JS(P||Q) = \frac{1}{2}(KL(P||M) + KL(Q||M)), \quad M = \frac{1}{2}(P+Q), \tag{1}$$

where *KL* represents the Kullback-Leibler divergence (Csiszár, 1975). A lower *JS* divergence value means that the test set is more similar to AMR2.0 training set on that specific feature.

As shown in Table 3, the main difference between the test sets and the training set comes from the input length, the unigram/bigram/trigram, the concept, and the triplet, while the relation difference is relatively small. The vocabulary differences (e.g., unigram OOV rate and concept OOV rate) are relatively small compared with feature distribution divergence. Among all the test

[1] https://github.com/bjascob/amrlib

[2] Please refer to appendix A.1 for detailed definitions.

[3] https://catalog.ldc.upenn.edu/LDC2017T10

[4] https://catalog.ldc.upenn.edu/LDC2020T02

[5] https://amr.isi.edu/download/amr-bank-struct-v1.6.txt

[6] https://amr.isi.edu/download/2016-03-14/amr-release-test-bio.txt

[7] https://github.com/IBM/AMR-annotations

Datasets	Input Features				Output Features		
	Avg. Len	Unigram	Bigram	Trigram	Concept	Relation	Triplet
AMR2.0 (ID)	19.76	0.14 (0.05)	0.46 (0.39)	0.64 (0.78)	0.14 (0.04)	0.07 (0.00)	0.43 (0.32)
New3	14.69	0.24 (0.10)	0.57 (0.50)	0.68 (0.85)	0.26 (0.10)	0.03 (0.00)	0.53 (0.44)
TLP	13.69	0.22 (0.04)	0.51 (0.34)	0.66 (0.78)	0.30 (0.05)	0.06 (**1e-3**)	0.60 (0.57)
Bio	**25.20**	**0.39** (**0.29**)	0.63 (**0.78**)	0.69 (**0.95**)	**0.44** (**0.21**)	0.06 (**1e-3**)	**0.66** (**0.71**)
QALD-9	7.52	0.38 (0.08)	**0.64** (0.48)	**0.69** (0.84)	0.38 (0.08)	**0.07** (0.00)	0.58 (0.40)

Table 3: Feature difference between AMR2.0 training set and 5 test sets. We report the Jensen-Shannon divergence (and OOV rate) for features except input length. A lower JS divergence (and OOV) value means that the test set is more similar to AMR2.0 training set on that specific feature.

Model	**ID**	**OOD**				
	AMR2.0	**New3**	**TLP**	**Bio**	**QALD-9**	**Avg**
JAMR	67.0	57.2 (14.6%)	59.9 (11.9%)	38.7 (42.2%)	60.8 (9.3%)	54.2 (19.2%)
AMRGS	80.6	61.8 (23.3%)	73.7 (9.4%)	43.9 (45.5%)	70.0 (13.1%)	62.4 (22.6%)
STRUCTBART	84.1	74.0 (12.0%)	80.2 (4.9%)	60.4 (28.2%)	83.7 (**0.5%**)	74.6 (11.3%)
SPRING	84.7	74.2 (12.2%)	79.9 (6.0%)	59.7 (29.5%)	80.4 (4.9%)	73.6 (13.2%)
AMRBART	**85.5**	**77.3** (**9.6%**)	**81.6** (**4.8%**)	**63.2** (**26.1%**)	**85.1** (**0.5%**)	**76.8** (**10.2%**)

Table 4: SMATCH scores on in-domain (ID) and out-of-domain (OOD) test sets and the relative performance reduction rate for OOD test sets. The best results within each column are shown in **bold**.

sets, AMR2.0 is the closest to the training set. In contrast, Bio has the longest average input length and the largest overall feature difference from the AMR2.0 training data. In particular, the unigram OOV rate of Bio is 0.29, which is much bigger than that of other test sets. New3 and TLP have medium input length, and the feature differences are smaller than Bio. QALD-9 has an average input length of 7.5, which is 2.5 times smaller than that of AMR2.0. Overall, we can observe that individual statistics vary across domains, which reflects large domain differences.

4.4 Cross-Domain AMR Parsing Performance

Table 4 lists the performances of 5 parsers on 5 domains. All models achieve their best results on the in-domain AMR2.0 test set. By contrast, the performance drops on OOD test sets, ranging from 0.5% to 45.5%, showing that cross-domain AMR parsing is still a challenge.

Among all domains, Bio is the hardest one, which is in line with our observation in feature differences (i.e., Table 3). The SMATCH scores of all parsers on Bio fall to a range between 38.7 and 63.2, which is much lower than those on ID test sets (from 67.0 to 85.5). The reason can be two-fold: First, Bio contains many biomedical terminologies, resulting in significant feature differences. For example, the unigram OOV rate is 29%, and the JS divergence of concept is 0.44 on Bio, according to Table 3. Second, the average input text length of Bio is larger than those of other test sets. In comparison, QALD-9 is the easiest, with a relative performance reduction rate ranging from 0.5% to 13.1%. The main reason could be that the input text length of QALD-9 is small, which significantly reduces the difficulty of AMR parsing. We give further analysis on these features in Section 5.

Among all the systems, AMRBART gives the best SMATCH scores on all test sets and has the lowest relative performance reduction rates, indicating that large-scale graph pre-training, which has been shown to boost ID performance, is also helpful for improving OOD generalization. SPRING gives better results than STRUCTBART on the ID test set and a bigger relative performance reduction rate on OOD test sets, which may result from the fact that the transition-based model implicitly learns the local correspondence between AMR and text, which is helpful for generalization. In contrast, the seq2seq-based model focus on sequence-level transduction. Comparing AMRGS with other models, though all four neural parsers achieve SMATCH scores of over 80.0, AMRGS shows much lower OOD performances than the other three neural parsers, and some of its OOD relative performance reduction rates are even bigger than those of the non-neural JAMR parser. This might be caused by the difference on rule-based processing methods for AMR graphs: as

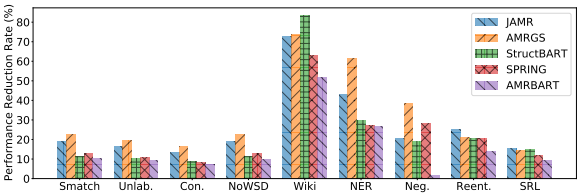


Figure 3: Relative performance reduction rate in terms of different evaluation metrics. "Unlab."–unlabeled, "Con."–Concept, "Neg"–Negation, "Reent."–Reentrancy.

shown in Table 1, AMRGS utilizes much more rule-based methods to pre- (and post-) process AMR graphs than the other three neural methods do. Since these rules are derived from the training data, such domain-specific rules would not generalize well to new domains. We also give the full evaluation results, please refer Appendix A.2.

5 Key Challenges to OOD AMR Parsing

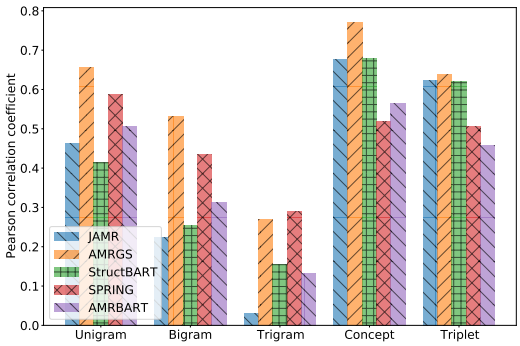
Based on the results above, we further study two important questions: *which AMR components are the most challenging for cross-domain AMR parsing (in Section 5.1)*; and *what contributes most to the performance degradation on OOD test sets (in Section 5.2)*?

5.1 Error Analysis

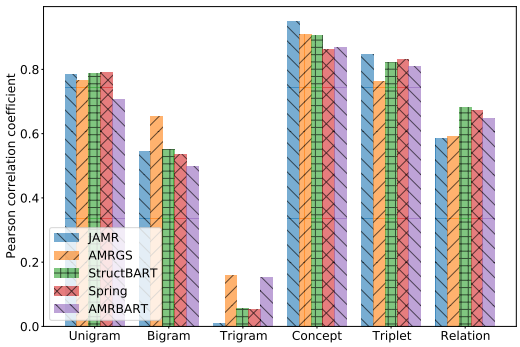
Figure 3 gives the relative performance reduction of each model regarding SMATCH and 8 fine-grained evaluation metrics (measured by F1 scores). We report the average score of four OOD test sets. Among all evaluation metrics, wiki links and named entities are the hardest objectives to handle. This is intuitive because a large proportion of wiki links and named entities contain out-of-vocabulary (OOV) tokens. Also, there are OOV named entity types in the OOD test sets. For example, 3.7% of named entity types of Bio are never seen during training, which further increases the difficulty level for AMR parsers to predict the correct labels. The performance of negative polarity and reentrancy detection also drops significantly, with average scores of 22% and 20%, respectively. For unlabeled[8], concept, NoWSD[9] and semantic role labeling (SRL), we observe relatively lower performance degradation.

[8] SMATCH while ignoring relation labels.

[9] SMATCH while ignoring Propbank senses.



(a)



(b)

Figure 4: Pearson correlation coefficient between performance (SMATCH) degradation rate and difference of feature distribution measured by (a) OOV rate; (b) Jensen-Shannon divergence. We do not include relation in sub-figure (a) because the relation OOV rate of most test sets are zeros.

5.2 Feature Analysis

To study the key factors that impact cross-domain AMR parsing performance, we measure the Pearson correlation coefficient between a set of linguistic features (as introduced in Section 4.3) and the relative performance degradation rate. To eliminate the influences of domain-specific features[10], we concatenate all OOD test sets and apply bootstrapping procedure (Efron and Tibshirani, 1994; Koehn, 2004) to obtain a number of simulated test sets, which are taken as samples for calculating the correlation scores. Specifically, we create 100 homologous test sets, each with 2,000 examples (out of 3,147) sampled from the concatenated set. We consider both Jensen-Shannon divergence and OOV rate to measure feature differences. In Figure 4, each group of columns shows the linear correlation coefficient

[10] For example, QALD-9 has the smallest averaged input length among all domains, so the relatively high OOD performance on QALD-9 does not imply that its concept / relation gives smaller domain shift than other domains do.

between the domain divergences of a specific feature (i.e., Table 3) and the cross-domain performance degradation rates of a specific parser (i.e., Table 4). We have the following observations:

- It can be observed that all parsers are more influenced by domain shift of uni-gram token features while less by those of more complex token features such as bi-gram and tri-gram. The reason might be that AMR parsers rely more on the particular token itself rather than its context for concept identification.

- Concepts have larger influences on parsers' performances than relations, indicating that concept identification is the main bottleneck for cross-domain AMR parsing.

- Compared with input textual features, all parsers are more influenced by output AMR structural features, which is consistent with findings of Yang et al. (2022) and Cui et al. (2022) in constituency parsing. . This suggest that future cross-domain AMR parsing systems should pay more attention on AMR structures.

6 Bridging the Domain Gap

According to our analysis in Section 5.2, we investigate two approaches to improve the model performance on OOD datasets by bridging the distribution gap between the training and test domains without modifying model structures.

Alleviating Input Feature Divergence. We employ raw text from target domain to enrich the model with domain-specific input features. Specifically, we collect raw text from the biomedical and fairy tales domain, which are then used as extra knowledge for training. Inspired by previous work (Gururangan et al., 2020), we add an intermediate pre-training step to adapt the pre-trained model to the target domain, which refers to domain-adaptive pre-training. We take BART (Lewis et al., 2020) as the backbone and continually pre-train BART on the collected dataset using the standard self-supervised learning training objective. We randomly mask text spans, replacing 15% tokens. The adaptively trained model is used for initialization during fine-tuning.

Alleviating Output Feature Divergence. We investigate silver data as pseudo target domain training data to fine-tune the AMR parsers. In this way, we expect the cross-domain distribution divergence of both text and AMR features can be reduced. We construct the silver data by applying a pre-trained AMR parser to parse collected domain-specific data into AMR graphs. We use a mixture of gold and silver data to train the models.

Model	Bio	TLP
With OOD raw data		
Unigram/Concept diver.	0.28/0.44	0.18/0.30
StructBART	**61.2** (+0.8)	**80.7** (+0.5)
SPRING	61.0 (+1.3)	80.4 (+0.5)
With OOD silver data		
Unigram (Concept) diver.	0.28 (0.30)	0.18 (0.22)
StructBART	62.8 (+2.4)	**81.3** (+1.1)
SPRING	**63.0** (+3.3)	81.1 (+1.2)

Table 5: SMATCH score (and improvements) on Bio and TLP when training with out-of-domain data. "diver."– Jensen-Shannon divergence.

Results. Table 5 shows the results of two BART-based systems[11] on Bio and TLP. First, with domain-specific raw data, the Jensen-Shannon divergence of unigram reduces significantly (p<0.01) compared with Table 3, reaching 0.28 and 0.18 on Bio and TLP, respectively. Both parsers give better results when initialized with the adaptively pre-trained model. This confirms our assumption that reducing the input distribution gap can benefit cross-domain AMR parsing. In addition, the distribution divergence of both unigram and concept decrease when using domain-specific silver data, and both models obtain significant improvements, with a large margin of 2.4 and 3.3 points on Bio. This suggests that reducing distribution divergence of AMR features can also lead to better results. Finally, compared with input textual features, AMR features give larger improvements. This is consistent with our observations in Section 5.2.

Fine-grained Evaluation. Figure 5 shows the fine-grained evaluation results on Bio. We take the original SPRING model (Original) as a baseline and compare the performance with the model augmented by domain adaptive pre-training (DAPT) and silver data (Silver). It can be observed that both methods improve the NER score over the baseline by a large margin (up to 6 points). Also, both methods give better results on concept identification, reentrancy detection, and semantic

[11]We do not consider AMRBART, because AMRBART has been trained using large-scale silver data.

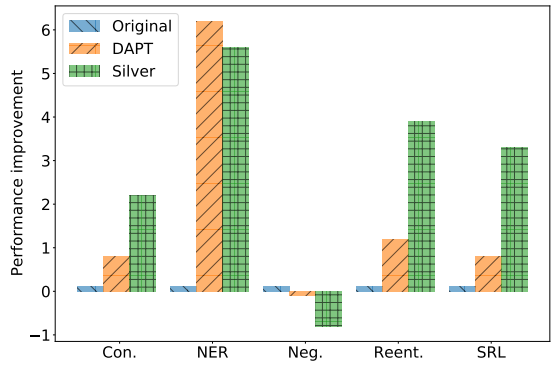


Figure 5: Performance improvements regarding fine-grained evaluation metrics. "Con."–Concept, "Neg."–Negation, "Reent."–Reentrancy.

role labeling. Compared with DAPT, Silver obtains significantly better results on graph-aware metrics (i.e., concept, reentrancy and SRL), showing that silver data can improve the model performance on predicting structures. The results of Silver is weaker than DAPT on text-related metrics (i.e., NER and negation). A possible reason is that silver data might contain noise, which hinders the model to make predictions from textual features.

7 Conclusion

We investigated the cross-domain generalization challenges for AMR parsing by analyzing the performance of five representative models. Empirically, we found that all AMR parsers give lower performance on OOD test sets, and the difficulty lies more in output features divergences, including concept and relation, compared with input features. Based on our analysis, we investigated two approaches to bridge the domain gap of input and output features, respectively, which achieve higher scores on out-of-domain test sets than previous work. In the future, we would like to investigate more methods, such as vocabulary adaptation (Sato et al., 2020) and k-nearest-neighbor (KNN, Khandelwal et al. 2020, 2021), to improve cross-domain AMR parsing.

Limitations

The limitation of our work can be stated from three perspectives. First, the proposed methods do not improve the in-domain parsing performance. Second, we only analyze the cross-domain performance of five representative AMR parsers. Third, we focus on cross-domain AMR parsing in one major language. The performance of other languages remains unknown.

Acknowledgments

Yue Zhang is the corresponding author. We would like to thank anonymous reviewers for their insightful comments and Yuchen Niu for his help in preliminary experiments. This work is supported by the National Natural Science Foundation of China under grant No.61976180 and the Tencent AI Lab Rhino-Bird Focused Research Program.

References

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022a. 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.

Xuefeng Bai, Linfeng Song, and Yue Zhang. 2020. Online back-parsing for AMR-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1206–1219, Online. Association for Computational Linguistics.

Xuefeng Bai, Linfeng Song, and Yue Zhang. 2022b. Semantic-based pre-training for dialogue understanding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 592–607, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation

for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. 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. 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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

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

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. Association for Computational Linguistics.

Deng Cai and Wai Lam. 2020a. 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.

Deng Cai and Wai Lam. 2020b. Graph transformer for graph-to-sequence learning. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7464–7471. AAAI Press.

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.

Kehai Chen, Muyun Yang, Tiejun Zhao, and Min Zhang. 2022a. Data-driven fuzzy target representation for intelligent translation system. *IEEE Transactions on Fuzzy Systems*, pages 1–1.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Yulong Chen, Yang Liu, Li Dong, Shuohang Wang, Chenguang Zhu, Michael Zeng, and Yue Zhang. 2022b. Adaprompt: Adaptive model training for prompt-based nlp. *arXiv preprint arXiv:2202.04824*.

Yulong Chen, Ming Zhong, Xuefeng Bai, Naihao Deng, Jing Li, Xianchao Zhu, and Yue Zhang. 2022c. The cross-lingual conversation summarization challenge. *arXiv preprint arXiv:2205.00379*.

Caitlin Christianson, Jason Duncan, and Boyan Onyshkevych. 2018. Overview of the darpa lorelei program. *Machine Translation*, 32(1–2):3–9.

Imre Csiszár. 1975. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pages 146–158.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.

Leyang Cui, Sen Yang, and Yue Zhang. 2022. Investigating non-local features for neural constituency parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2065–2075, Dublin, Ireland. Association for Computational Linguistics.

Daniel Dahlmeier and Hwee Tou Ng. 2010. Domain adaptation for semantic role labeling in the biomedical domain. *Bioinformatics*, 26(8):1098–1104.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Quynh Thi Ngoc Do, Steven Bethard, and Marie-Francine Moens. 2015. Domain adaptation in semantic role labeling using a neural language model and linguistic resources. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(11):1812–1823.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

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. Association for Computational Linguistics.

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

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

B. Fuglede and F. Topsoe. 2004. Jensen-shannon divergence and hilbert space embedding. In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, pages 31–.

Zhijiang Guo and Wei Lu. 2018. Better transition-based AMR parsing with a refined search space. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1712–1722, Brussels, Belgium. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Pengcheng He, Baolin Peng, Liyang Lu, Song Wang, Jie Mei, Yang Liu, Ruochen Xu, Hany Hassan Awadalla, Yu Shi, Chenguang Zhu, et al. 2022. Z-code++: A pre-trained language model optimized for abstractive summarization. *arXiv preprint arXiv:2208.09770*.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1138, Vancouver, Canada. Association for Computational Linguistics.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2018. Multitask parsing across semantic representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 373–385, Melbourne, Australia. 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. Association for Computational Linguistics.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. 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 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. 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. Association for Computational Linguistics.

Young-Suk Lee, Ramon Fernandez Astudillo, Thanh Lam Hoang, Tahira Naseem̃, Radu Florian, and Salim Roukos. 2022. Maximum bayes smatch ensemble distillation for AMR parsing. In *NAACL-HLT 2022: The 2022 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies*.

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.

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. Association for Computational Linguistics.

Soojong Lim, Changki Lee, Pum-Mo Ryu, Hyunki Kim, Sang Kyu Park, and Dongyul Ra. 2014. Domain-adaptation technique for semantic role labeling with structural learning. *ETRI Journal*, 36(3):429–438.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, Denver, Colorado. 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. Association for Computational Linguistics.

Yun Luo, Hongjie Cai, Linyi Yang, Yanxia Qin, Rui Xia, and Yue Zhang. 2022. Challenges for open-domain targeted sentiment analysis. *arXiv preprint arXiv:2204.06893*.

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. Association for Computational Linguistics.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Qi Zhang, and Xuanjing Huang. 2021. Template-free prompt tuning for few-shot NER. *CoRR*, abs/2109.13532.

José-Lázaro Martínez-Rodríguez, Aidan Hogan, and Ivan López-Arévalo. 2020. Information extraction meets the semantic web: A survey. *Semantic Web*, 11(2):255–335.

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. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Zdeňka Urešová. 2016. Towards comparability of linguistic graph Banks for semantic parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3991–3995, Portorož, Slovenia. European Language Resources Association (ELRA).

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. Association for Computational Linguistics.

Dheeraj Rajagopal, Nidhi Vyas, Aditya Siddhant, Anirudha Rayasam, Niket Tandon, and Eduard Hovy. 2019. Domain adaptation of SRL systems for biological processes. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 80–87, Florence, Italy. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227, Online. Association for Computational Linguistics.

Shoetsu Sato, Jin Sakuma, Naoki Yoshinaga, Masashi Toyoda, and Masaru Kitsuregawa. 2020. Vocabulary adaptation for domain adaptation in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4269–4279, Online. Association for Computational Linguistics.

Aviv Slobodkin, Leshem Choshen, and Omri Abend. 2022. Semantics-aware attention improves neural machine translation. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 28–43, Seattle, Washington. 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.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Rik Van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *arXiv preprint arXiv:1705.09980*.

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. 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. Association for Computational Linguistics.

Jing Wang, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2020. Multi-domain named entity recognition with genre-aware and agnostic inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8476–8488, Online. Association for Computational Linguistics.

Qifan Wang, Yi Fang, Anirudh Ravula, Fuli Feng, Xiaojun Quan, and Dongfang Liu. 2022. Webformer: The web-page transformer for structure information extraction. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 3124–3133, New York, NY, USA. Association for Computing Machinery.

Sam Wiseman and Karl Stratos. 2019. Label-agnostic sequence labeling by copying nearest neighbors. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5363–5369, Florence, Italy. Association for Computational Linguistics.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Sen Yang, Leyang Cui, Ruoxi Ning, Di Wu, and Yue Zhang. 2022. Challenges to open-domain constituency parsing. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 112–127, Dublin, Ireland. Association for Computational Linguistics.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online. Association for Computational Linguistics.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks.

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. 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. 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. 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. 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. Association for Computational Linguistics.

Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019. Modeling graph structure in transformer for better AMR-to-text generation. 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 5459–5468, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 Fine-grained Evaluation Metric for AMR Parsing

The Smatch score (Cai and Knight, 2013) measures the degree of overlap between the gold and the prediction AMR graphs. It can be further broken into different sub-metrics, including:

- Unlabeled (Unlab.): Smatch score after removing edge-labels
- NoWSD: Smatch score after ignoring Propbank senses (*e.g.*, go-01 vs go-02)
- Concepts (Con.): 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 (Reen.): 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.

A.2 Full Cross-domain Performance

Table 6 shows the detailed results of AMR parsing on different test sets in terms of 9 evaluation metrics.

Model	Smatch	Unlab.	NoWSD	Con.	Wiki.	NER	Reent.	Neg.	SRL
AMR2.0 (ID)									
JAMR	67.0	71.6	67.7	83.0	75.9	80.3	61.0	43.9	59.7
AMRGS	80.6	83.9	81.0	88.1	86.5	81.1	64.7	78.5	74.3
StructBART	84.1	87.6	84.4	90.4	79.6	**92.2**	**74.3**	71.2	**83.0**
SPRING	84.7	87.6	84.9	90.2	**87.3**	83.7	72.3	**79.9**	79.7
AMRBART	**85.5**	**88.4**	**85.9**	**91.2**	84.4	91.5	73.5	73.5	81.5
New3 (OOD)									
JAMR	57.2	62.5	57.8	73.1	49.8	52.7	38.9	28.3	53.2
AMRGS	61.8	66.8	62.2	75.9	49.6	45.4	54.8	59.6	65.0
StructBART	74.0	78.1	74.5	83.1	53.6	71.1	63.2	63.3	72.1
SPRING	74.2	78.4	74.6	82.3	60.1	66.4	62.9	64.2	71.7
AMRBART	**77.3**	**81.2**	**77.8**	**84.6**	**73.5**	**72.0**	**65.6**	**66.7**	**73.7**
TLP *v1.6* (OOD)									
JAMR	59.9	66.7	60.9	88	25.5	53.0	32.4	55.4	54.6
AMRGS	73.7	78.4	74.6	82.4	33.1	24.1	58.7	63.5	70.8
StructBART	80.2	84.3	81.0	87.1	69.5	**75.2**	67.9	77.3	77.4
SPRING	79.9	83.9	80.7	86.4	65.7	63.2	67.0	**80.9**	77.0
AMRBART	**81.6**	**85.3**	**82.3**	**87.8**	**87.4**	73.5	**69.3**	77.8	**78.7**
TLP *v3.0* (OOD)									
JAMR	58.8	66.0	59.7	75.9	25.5	53.0	31.7	49.1	52.9
AMRGS	72.0	77.0	72.9	81.3	33.1	24.1	57.2	57.5	68.5
StructBART	78.5	83.0	79.2	85.9	69.5	**75.2**	66.1	70.1	75.0
SPRING	78.2	82.6	79.0	85.3	65.7	63.2	65.0	**72.9**	74.7
AMRBART	**79.8**	**84.0**	**80.5**	**86.7**	**84.7**	73.5	**67.6**	70.7	**76.4**
Bio *v0.8* (OOD)									
JAMR	38.7	44.1	39.6	56.9	7.6	15.6	26.0	50.3	37.3
AMRGS	43.9	49.8	44.4	55.6	**9.0**	6.4	34.1	60.4	47.0
StructBART	60.4	64.9	60.9	70.1	1.9	31.9	43.6	76.0	56.7
SPRING	59.7	63.7	60.2	71.1	3.2	33.7	43.5	**75.7**	57.5
AMRBART	**63.2**	**67.2**	**63.9**	**73.4**	2.0	**39.7**	**47.1**	75.4	**60.6**
Bio *v3.0* (OOD)									
JAMR	38.4	43.9	39.3	56.7	7.5	15.6	25.8	44.9	36.7
AMRGS	43.2	49.1	43.7	55.0	**8.6**	6.4	33.8	55.6	45.9
StructBART	57.6	62.0	58.2	69.2	1.9	31.9	42.9	70.1	55.2
SPRING	59.2	63.5	59.6	70.2	3.2	33.7	43.2	**72.4**	56.2
AMRBART	**62.1**	**66.3**	**62.9**	**72.7**	2.0	**39.7**	**46.7**	70.5	**59.2**
QALD-9 (OOD)									
JAMR	60.8	66.6	61.3	69.9	0	61.2	34.2	5.0	56.4
AMRGS	70.0	74.5	70.2	80.1	0	48.7	57.0	9.5	72.0
StructBART	83.7	86.9	83.9	89.7	0	81.9	61.3	14.0	76.3
SPRING	80.4	83.2	80.6	88.9	0	80.4	56.5	9.3	75.4
AMRBART	**85.1**	**87.3**	**85.2**	**91.8**	0	**83.8**	**71.9**	**69.1**	**83.6**

Table 6: AMR parsing results on in-domain and out-of-domain test sets. The best results within each row block are shown in bold.