XLPT-AMR: Cross-Lingual Pre-Training via Multi-Task Learning for Zero-Shot AMR Parsing and Text Generation

Dongqin Xu[1] Junhui Li[1]* Muhua Zhu[2] Min Zhang[1] Guodong Zhou[1]
[1]School of Computer Science and Technology, Soochow University, Suzhou, China
[2]Tencent News, Beijing, China
xdqck@live.com, {lijunhui, minzhang, gdzhou}@suda.edu.cn
zhumuhua@gmail.com

Abstract

Due to the scarcity of annotated data, Abstract Meaning Representation (AMR) research is relatively limited and challenging for languages other than English. Upon the availability of English AMR dataset and English-to-X parallel datasets, in this paper we propose a novel cross-lingual pre-training approach via multi-task learning (MTL) for both zero-shot AMR parsing and AMR-to-text generation. Specifically, we consider three types of relevant tasks, including AMR parsing, AMR-to-text generation, and machine translation. We hope that knowledge gained while learning for English AMR parsing and text generation can be transferred to the counterparts of other languages. With properly pre-trained models, we explore four different fine-tuning methods, i.e., vanilla fine-tuning with a single task, one-for-all MTL fine-tuning, targeted MTL fine-tuning, and teacher-student-based MTL fine-tuning. Experimental results on AMR parsing and text generation of multiple non-English languages demonstrate that our approach significantly outperforms a strong baseline of pre-training approach, and greatly advances the state of the art. In detail, on LDC2020T07 we have achieved 70.45%, 71.76%, and 70.80% in Smatch F1 for AMR parsing of German, Spanish, and Italian, respectively, while for AMR-to-text generation of the languages, we have obtained 25.69, 31.36, and 28.42 in BLEU respectively. We make our code available on github https://github.com/xdqkid/XLPT-AMR.

1 Introduction

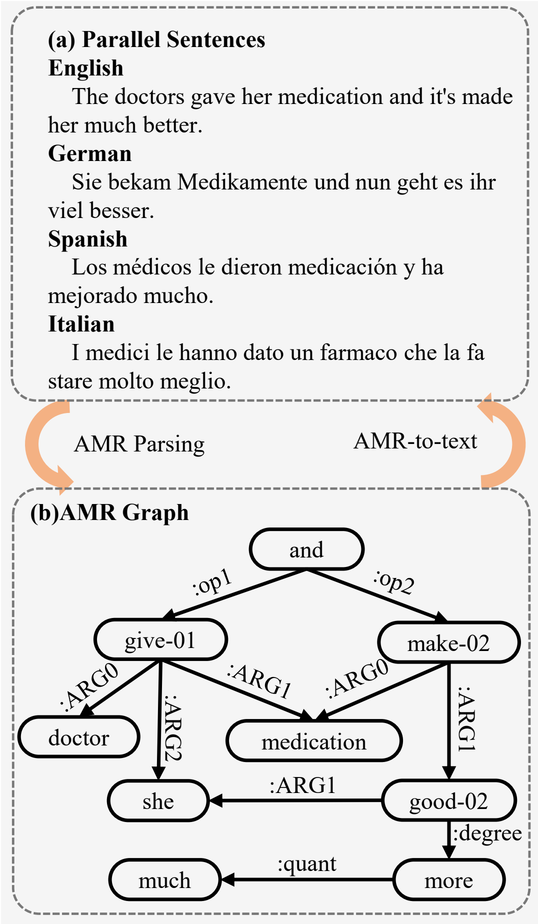
Abstract Meaning Representation (AMR) (Banarescu et al., 2013) is a widely used formalism that represents the semantics of a sentence with a directed and acyclic graph. Figure 1 (b) shows an example AMR graph where the nodes such as "doctor" and "give-01" represent concepts, and the edges such as ":ARG0" and ":ARG1" stand for semantic relations between two connected concepts. Recent studies on AMR mainly fall in two directions: AMR parsing which converts a sentence into an AMR graph (Flanigan et al., 2014; Wang et al., 2015a; Konstas et al., 2017, to name a few) and its inverse, i.e., AMR-to-text generation that produces a sentence from an AMR graph (Flanigan et al., 2016; Song et al., 2017, 2018, to name a few).

Restricted by the availability of annotated corpora, most of previous studies on AMR focus on English while very few studies are for Chinese and Portuguese (Wang et al., 2018; Sobrevilla Cabezudo et al., 2019; Anchiêta and Pardo, 2020). Cross-lingual AMR research, however, has received relatively less attention. In fact, cross-lingual AMR has mainly been studied in the scope of annotation works (Xue et al., 2014; Hajič et al., 2014). Till recently, Damonte and Cohen (2018) demonstrate that AMR annotated for English can be used as cross-lingual semantic representations, and propose to conduct cross-lingual AMR parsing via annotation projection and machine translation. Blloshmi et al. (2020) follow the same line and create large-scale silver data to boost the performance of cross-lingual AMR parsing. Fan and Gardent (2020) focus on multilingual AMR-to-text generation for twenty one different languages. The aforementioned studies consider AMR parsing and AMR-to-text generation separately.

In this paper, we formalize both AMR parsing and AMR-to-text generation as sequence-to-sequence (seq2seq) learning and propose a novel and effective approach to cross-lingual AMR, which is illustrated in Figure 1. Upon the availability of the English AMR dataset and English-to-X parallel datasets ($X \in \{\text{German}, \text{Spanish}, \text{Italian}\}$ in this paper), our purpose is to boost the performance of zero-shot AMR parsing and text generation in

[*] Corresponding Author: Junhui Li.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 896–907
August 1–6, 2021. ©2021 Association for Computational Linguistics



(a) Parallel Sentences

English

The doctors gave her medication and it's made her much better.

German

Sie bekam Medikamente und nun geht es ihr viel besser.

Spanish

Los médicos le dieron medicación y ha mejorado mucho.

Italian

I medici le hanno dato un farmaco che la fa stare molto meglio.

Figure 1: Illustration of cross-lingual AMR parsing and AMR-to-text generation: (a) sentences in different languages sharing the same meaning; (b) AMR graph of the sentences.

X-language. To this end, we borrow the idea of joint pre-training from Xu et al. (2020) and explore three types of relevant tasks, including machine translation tasks, AMR parsing and AMR-to-text generation tasks. We conjecture that knowledge gained while learning for English AMR parsing and text generation could be helpful to the X-language counterparts, and machine translation tasks could act as a good regularizer (Xu et al., 2020). To the best of our knowledge, this is the first study that utilizes such a pre-training approach in cross-lingual AMR research.

We also explore and compare four different fine-tuning methods to answer the question that whether combining AMR parsing and AMR-to-text generation tasks in fine-tuning stage will achieve better performance. Moreover, inspired by the teacher-student mechanism (Kim and Rush, 2016; Chen et al., 2017), we extend the fine-tuning method to improve a target fine-tuning task with the help of another relevant yet stronger task. Experimental results on the cross-lingual AMR dataset (LDC2020T07) show that the proposed approach greatly advances the state of the art of cross-lingual AMR.

Overall, we make the following contributions.

- We propose an effective cross-lingual pre-training approach for zero-shot AMR parsing and AMR-to-text generation. Our pre-trained models could be used for both AMR parsing and AMR-to-text generation.
- We explore and compare different fine-tuning methods. We also propose a teacher-student-based fine-tuning method that achieves the best performance.
- We evaluate our approach in three zero-shot languages of AMR and our approach greatly advances the state of the art.

2 Related Work

We describe related studies on AMR from three perspectives: English AMR parsing, English AMR-to-text generation, and cross-lingual AMR.

English AMR Parsing. AMR parsing is a task that translates a sentence into a directed and acyclic graph (Banarescu et al., 2013). According to the approaches to modeling the structure in AMR graphs, previous studies on AMR Parsing for English can be broadly grouped into several categories, which are tree-based approaches (Wang et al., 2015b; Groschwitz et al., 2018), graph-based approaches (Flanigan et al., 2014; Werling et al., 2015; Cai and Lam, 2019), transition-based approaches (Zhou et al., 2016; Damonte et al., 2017; Ballesteros and Al-Onaizan, 2017; Guo and Lu, 2018; Zhou et al., 2021), sequence-to-sequence (seq2seq) approaches (Peng et al., 2017; van Noord and Bos, 2017; Konstas et al., 2017; Ge et al., 2019; Xu et al., 2020; Bevilacqua et al., 2021), and sequence-to-graph (seq2graph) approaches (Lyu and Titov, 2018; Zhang et al., 2019a,b; Cai and Lam, 2020a).

English AMR-to-Text Generation. As an inverse task of AMR parsing, AMR-to-text generation aims to write a sentence from an AMR graph. Early studies on this task rely on grammar-based approaches (Flanigan et al., 2016; Song et al., 2017). More recent studies propose to regard AMR-to-text generation as a machine translation or seq2seq task (Pourdamghani et al., 2016; Ferreira et al., 2017; Konstas et al., 2017; Cao and Clark, 2019). However, seq2seq approaches tend to lose structural information in AMR graphs since they simply linearize AMR graphs into sequences before feeding them into the models. To prevent information loss caused by linearization, a variety of graph-to-sequence approaches have been proposed to better model structural information (Song et al., 2018; Beck et al., 2018; Damonte and Cohen, 2019; Guo et al., 2019; Ribeiro et al., 2019; Zhu et al., 2019; Cai and Lam, 2020b; Zhao et al., 2020; Song et al., 2020; Yao et al., 2020; Bai et al., 2020). By taking advantages of strong pre-trained language models, recent studies achieve new state of the art (Mager et al., 2020; Harkous et al., 2020; Ribeiro et al., 2020; Bevilacqua et al., 2021) .

Cross-Lingual AMR. All above related studies focus on English AMR research. Relatively limited efforts have been put on other languages due to the lack of language-specific AMR corpora. Actually, whether AMR can act as an interlingua is an open question (Xue et al., 2014; Hajič et al., 2014). Till lately , Damonte and Cohen (2018) demonstrate that a simplified AMR can be used across languages and for the first time they study cross-lingual AMR parsing for languages rather than English. Blloshmi et al. (2020) employ large-scale silver parallel AMR data to bridge the gap between different languages and greatly advance the performance of cross-lingual AMR parsing. Sheth et al. (2021) explore annotation projection to leverage existing English AMR and overcome resource shortage in the target language. Furthermore, Fan and Gardent (2020) explore cross-lingual AMR-to-text based on pre-trained cross-lingual language model (XLM) (Lample and Conneau, 2019). In this paper we build strong cross-lingual pre-trained models for both AMR parsing and AMR-to-text generation. Moreover, a nice property of our approach is that for AMR parsing, unlike related studies (Damonte and Cohen, 2018; Blloshmi et al., 2020), we do not need to perform lemmatization, POS tagging, NER, or re-categorization of entities, thus require no language specific toolkits in pre-processing.

3 Cross-Lingual Pre-Training

In this section, we first present the background of our pre-training approach (Section 3.1), followed by the description of cross-lingual pre-training tasks (Section 3.2). Then we present our joint pre-training (Section 3.3). For simplicity, in the following we use German as a representative to describe our approach to German AMR parsing and AMR-to-text generation.

3.1 Background

Transformer-based Seq2Seq Learning. Our models are built on the Transformer framework (Vaswani et al., 2017). The encoder in Transformer consists of a stack of multiple identical layers, each of which has two sub-layers: one implements the multi-head self-attention mechanism and the other is a position-wise fully-connected feed-forward network. The decoder is also composed of a stack of multiple identical layers. Each layer in the decoder consists of the same sub-layers as in the encoder plus an additional sub-layer that performs multi-head attention to the distributional representation produced by the encoder. See Vaswani et al. (2017) for more details.

AMR Graph Linearization and Recovering. To make Transformer applicable to AMR parsing and AMR-to-text generation, on the one hand we follow van Noord and Bos (2017) to linearize AMR graphs into sequences by removing variables, wiki links and duplicating the co-referring nodes. On the other hand, for AMR parsing we need to recover the graph representation from linearized AMRs by assigning a unique variable to each concept, pruning duplicated and redundant materials, restoring co-referring nodes, fixing incomplete concepts and performing Wikification.[1] In this paper, we adopt linearization and recovering scripts provided by van Noord and Bos (2017).[2]

3.2 Cross-Lingual Pre-Training Tasks

Due to the unavailability of gold training data of German AMR parsing and AMR-to-text generation, we view English as a pivot and hope that knowledge gained while learning for English AMR parsing and text generation could be helpful for the German counterparts. Specifically, given an EN-DE parallel dataset $(\mathcal{T}^{EN}, \mathcal{T}^{DE})$, we use an English AMR parser trained on annotated English AMRs (i.e., AMR2.0) to parse the English sentences into AMR graphs, thus obtain a trilingual parallel dataset $\mathcal{T} = (\mathcal{T}^{EN}, \mathcal{T}^{DE}, \mathcal{T}^{AMR})$. Then

[1] We extract a term-wiki list from English AMR training dataset. When performing Wikification, we simply just look up the list.

[2] https://github.com/RikVN/AMR

on the trilingual parallel dataset, we propose cross-lingual pre-training via multi-task learning. We consider three types of tasks, i.e., AMR parsing, AMR-to-text generation, and machine translation.

AMR Parsing Tasks, which include both English AMR parsing on the training data $\left(\mathcal{T}^{EN}, \mathcal{T}^{AMR}\right)$ and German AMR parsing on $\left(\mathcal{T}^{DE}, \mathcal{T}^{AMR}\right)$. Note that both AMR parsing tasks are trained on silver AMR graphs.

AMR-to-Text Generation Tasks, which include both English AMR-to-text generation and German AMR-to-text generation. Similar to AMR parsing, these two AMR-to-text generation tasks are also trained on silver AMR graphs $\left(\mathcal{T}^{AMR}, \mathcal{T}^{EN}\right)$ and $\left(\mathcal{T}^{AMR}, \mathcal{T}^{DE}\right)$, respectively.

Machine Translation Tasks, which include both English-to-German and German-to-English machine translation tasks on $\left(\mathcal{T}^{EN}, \mathcal{T}^{DE}\right)$. The advantage of including the bi-directional translation tasks is three-fold. First, English-to-German translation will enable the decoder to generate fluent German sentence, which is beneficial to German AMR-to-text generation. Second, German-to-English translation will enable the encoder to capture syntax and semantic information from German sentences, which is beneficial to German AMR parsing. Third, translation tasks can serve as regularization to the training of AMR parsing and AMR-to-text generation, both of which are apt to overfit to the training data.

Overall speaking, in our pre-training there exist three types of (six) pre-training tasks in total. The pre-training is conducted on a trilingual parallel dataset $\left(\mathcal{T}^{EN}, \mathcal{T}^{DE}, \mathcal{T}^{AMR}\right)$, where $\mathcal{T}^{EN}$ and $\mathcal{T}^{DE}$ are parallel gold sentence pairs while $\mathcal{T}^{AMR}$ is the set of corresponding silver AMR graphs.

3.3 Jointly MTL Pre-Training

To train the above six pre-training tasks with a single model, we follow the strategy used in Xu et al. (2020) and add preceding language tags to both source and target sides of training data to distinguish the inputs and outputs of each training task. As illustrated in Table 1, we use <en>, <de>, and <amr> as the tags of begin-of-sentence for English sentences, German sentences, and linearized AMRs, respectively.

Our joint pre-training on multiple tasks falls into the paradigm of multi-task learning (MTL). In the training stage, we take turns to load the training data of these pre-training tasks. For example, we update model parameters on a batch of training instances from the first task, and then update parameters on a batch of training instances of the second task, and the process repeats. We also note that, according to our preliminary experimentation, the effect of different orders of carrying out these pre-training tasks is negligible.

English	<en> English Sentence
German	<de> German Sentence
AMR	<amr> Linearized AMR

Table 1: Preceding tags as the symbol of begin-of-sentence to distinguish languages.

4 Fine-Tuning Methods

To fine-tune a pre-trained model, we create a fine-tuning dataset from English annotated AMRs (i.e.,AMR2.0). Given English-AMR parallel data $\left(\mathcal{F}^{EN}, \mathcal{F}^{AMR}\right)$, we use an English-to-German translator to translate the English sentences into German sentences, thus obtain trilingual parallel dataset $\mathcal{F} = \left(\mathcal{F}^{EN}, \mathcal{F}^{DE}, \mathcal{F}^{AMR}\right)$. As our goal is to improve the performance of zero-shot AMR parsing and AMR-to-text generation, our primary fine-tuning tasks are German AMR parsing and AMR-to-text generation. Moreover, we could include the other four fine-tuning tasks as auxiliary tasks when necessary, i.e., English AMR parsing and AMR-to-text generation, as well as English-to-German and German-to-English translation.

Once the fine-tuning dataset is ready, we can fine-tune a pre-trained model with different methods. The vanilla fine-tuning method that fine-tunes a pre-trained model on the dataset of a primary task is a natural choice. We can also fine-tune a pre-trained model jointly over all fine-tuning tasks, or over the primary tasks plus specifically chosen fine-tuning tasks that are relevant. In the following we explore and compare four different fine-tuning methods.

4.1 Vanilla Fine-Tuning

Given a pre-trained model, vanilla fine-tuning updates the parameters of the pre-trained model solely on the dataset of the downstream task. For example, for German AMR parsing, we fine-tune the pre-trained model on the fine-tuning dataset of the German AMR parsing task. In other words, vanilla fine-tuning involves only a single-task learning.

4.2 One-for-All MTL Fine-Tuning

We fine-tune a pre-trained model synchronously for all six fine-tuning tasks, which are the same as the pre-training tasks. Related studies (Li and Hoiem, 2018; Xu et al., 2020) have shown that it is important to optimize for high accuracy of a primary fine-tuning task while preserving the performance of other tasks. Preserving the performance of various pre-training tasks could be viewed as a regularizer for each fine-tuning task. Similarly to joint pre-training, we take turns to load the fine-tuning data of these fine-tuning tasks. Consequently, we obtain a single fine-tuned model for all tasks.

4.3 Targeted MTL Fine-Tuning

Rather than including all fine-tuning tasks within a single model, we can selectively choose relevant fine-tuning tasks. For German AMR parsing, we use AMR parsing on German as the primary fine-tuning task and German-to-English translation as an auxiliary fine-tuning task. The auxiliary task will enhance the encoder to capture semantic information from German sentences. This is also consistent with the fine-tuning tasks designed for English AMR parsing in (Xu et al., 2020). For German AMR-to-text generation, we choose English-to-German as the auxiliary fine-tuning task, which is beneficial for the decoder to generate fluent German sentences.

4.4 Teacher-Student-based MTL Fine-Tuning

One notable property of the fine-tuning dataset is that the German sentences are produced automatically through machine translation. Noises in such silver fine-tuning dataset may degrade the performance of fine-tuned models. Inspired by the teacher-student framework (Kim and Rush, 2016; Chen et al., 2017), we propose to solve this problem by using a stronger fine-tuning task to help improve fine-tuning tasks on such noisy data. For example, we can use English AMR parsing (as the teacher) to help German AMR parsing (as the student), since English AMR parsing that is fine-tuned on gold data tends to have stronger performance.

Fine-Tuning for German AMR Parsing. We use E, G, A to denote English-side, German-side, and AMR-side, respectively, and $(\mathbf{e}, \mathbf{g}, \mathbf{a})$ as a triple instance. For German AMR parsing (i.e., $G \to A$), we regard English AMR parsing (i.e., $E \to A$) as its teacher and assume that the probability of generating a target AMR token a_i from $\mathbf{g}$ should be close to that from its counterpart $\mathbf{e}$, given the already obtained partial AMR $\mathbf{a}_{<i}$. On this assumption, the student model can acquire knowledge from the teacher by applying word-level knowledge distillation for multi-class cross-entropy with the following joint training objective:

$$\mathcal{J}\left(\theta_{G\to A}\right) = \sum_{(\mathbf{e},\mathbf{g},\mathbf{a})} J\left(\mathbf{e},\mathbf{g},\mathbf{a},\hat{\theta}_{E\to A},\theta_{G\to A}\right) + L_{\theta_{G\to A}}\left(\mathbf{a}\mid\mathbf{g}\right), \quad (1)$$

where $(\mathbf{e},\mathbf{g},\mathbf{a}) \in D_{E,G,A}$, i.e., $(\mathcal{F}^{EN}, \mathcal{F}^{DE}, \mathcal{F}^{AMR})$, the fine-tuning data for English/German AMR parsing, $\hat{\theta}_{E\to A}$ denotes the already learned model parameters for English AMR parsing,[3] and $L_{\theta_{G\to A}}\left(\mathbf{a}\mid\mathbf{g}\right)$ denotes the log-likelihood function for *translating* $\mathbf{g}$ into $\mathbf{a}$. The function J in Eq. 1 is defined as:

$$\begin{aligned} & J\left(\mathbf{e},\mathbf{g},\mathbf{a},\hat{\theta}_{E\to A},\theta_{G\to A}\right) \\ & = \sum_{i=1}^{|\mathbf{a}|} \mathrm{KL}\left(P(a|\mathbf{e},\mathbf{a}_{<i};\hat{\theta}_{E\to A}) \parallel P(a|\mathbf{g},\mathbf{a}_{<i};\theta_{G\to A})\right) \\ & = \sum_{i=1}^{|\mathbf{a}|} \sum_{a\in\mathcal{V}_{\mathbf{a}}} P(a|\mathbf{e},\mathbf{a}_{<i};\hat{\theta}_{E\to A}) \log \frac{P(a|\mathbf{e},\mathbf{a}_{<i};\hat{\theta}_{E\to A})}{P(a|\mathbf{g},\mathbf{a}_{<i};\theta_{G\to A})}, \end{aligned} \quad (2)$$

where $\mathrm{KL}(\cdot \parallel \cdot)$ denotes the KL divergence between two distributions, and $\mathcal{V}_{\mathbf{a}}$ is the vocabulary set.[4]

To sum up, in MTL fine-tuning we use Eq. 1 as the objective for the fine-tuning task of German AMR parsing while we still use the log-likelihood function for the auxiliary fine-tuning task, i.e., German-to-English translation.

Fine-Tuning for German AMR-to-Text Generation. Considering the fact that the performance of English-to-German translation is also better than that of German AMR-to-text generation, we view English-to-German translation as the teacher and assume that the probability of generating a target German token g_i from $\mathbf{a}$ should be close to that from its counterpart $\mathbf{e}$, given the already obtained partial German sentence $\mathbf{g}_{<i}$. The joint training objective for German AMR-to-text generation is similar to the aforementioned objective function for German AMR parsing. Due to limited space, we omit definition details of the objective function.

[3] The English AMR parser is learned by fine-tuning the pre-trained model on fine-tuning tasks of English AMR parsing and English-to-German translation.

[4] To avoid overfitting, the method additionally fine-tunes 80K steps on the pre-training dataset at the beginning.

5 Experimentation

In this section, we report the performance of our approach to AMR parsing and AMR-to-text generation for non-English languages, including German (DE), Spanish (ES), and Italian (IT). The models are pre-trained and fine-tuned on English data and one of either DE, ES, or IT, and are evaluated in the target language.

5.1 Experimental Settings

Pre-Training Datasets. For German, we use the WMT14 English-German translation dataset [5] which consists of 3.9M sentence pairs after preprocessing. For Spanish and Italian, we use Europarl parallel datasets,[6] which consist of 1.9M English-Spanish and 1.9M English-Italian sentence pairs, respectively. The English sentences of all the datasets are all parsed into AMR graphs via an English AMR parser trained on AMR 2.0 (LDC2017T10) (Appendix A provides more details on the English AMR parser). We merge English, German (Spanish/Italian) sentences and linearized AMRs together and segment all the tokens into subwords by byte pair encoding (BPE) (Sennrich et al., 2016) with 40K (or 30K for both Spanish and Italian) operations.

In addition, we also train NMT models to translate English into German, Spanish, and Italian on above parallel datasets with Transformer-big settings (Vaswani et al., 2017). These NMT models will be used in preparing fine-tuning datasets (Appendix B provides more implementation details on the NMT models).

Fine-Tuning Datasets. We use English AMR2.0 which contains 36,521, 1,368, and 1,371 English-AMR pairs for training, development, and testing, respectively. We translate the English sentences into German, Spanish, and Italian, respectively. We segment all the tokens into subwords by using the BPE model trained on pre-training datasets.

Pre-Training and Fine-Tuning Model Settings. We implement above pre-trained models based on *OpenNMT-py* (Klein et al., 2017). [7] For simplicity, we use the same hyperparameter settings to train all the models in both pre-training and fine-tuning by just following the settings for the Transformer-base model in Vaswani et al. (2017). The number of layers in encoder and decoder is 6 while the number of heads is 8. Both the embedding size and the hidden state size are 512 while the size of feedforward network is 2048. Moreover, we use Adam optimizer (Kingma and Ba, 2015) with β_1 of 0.9 and β_2 of 0.98. Warm_up step, learning rate, dropout rate, and label smoothing epsilon are set to 16000, 2.0, 0.1 and 0.1 respectively. We set the batch size to 4,096 (8,196) in pre-training (fine-tuning). We pre-train (fine-tune) the models for 250K (10K) steps and save them at every 10K (1K) steps. Finally, we obtain final pre-trained (fine-tuned) models by averaging the last 10 checkpoints.

Evaluation. We evaluate on LDC2020T07 (Damonte and Cohen, 2018), a corpus containing human translations of the test portion of 1371 sentences from the AMR 2.0, in German, Spanish, Italian, and Chinese. This data is designed for use in cross-lingual AMR research. Following Fan and Gardent (2020), we only evaluate on languages of German, Spanish and Italian where we have training data from EUROPARL. For AMR parsing evaluation, we utilize Smatch and other fine-grained metrics (Cai and Knight, 2013; Damonte et al., 2017). For AMR-to-text generation, we report performance in BLEU (Papineni et al., 2002).

5.2 Baseline Systems

We compare the performance of our approach against two baseline systems.

Baseline$_{\text{scratch}}$. To build this baseline system, we directly train models from scratch on the fine-tuning datasets. Taking German AMR parsing as example, we train the model on its fine-tuning dataset $\left(\mathcal{F}^{\text{DE}}, \mathcal{F}^{\text{AMR}}\right)$ to get Baseline$_{\text{scratch}}$.

Baseline$_{\text{pre-trained}}$. Rather than training models from scratch, we pre-train the models on large-scale silver datasets. Taking German AMR parsing as example, we first pre-train the model on the pre-training dataset, i.e., $\left(\mathcal{T}^{\text{DE}}, \mathcal{T}^{\text{AMR}}\right)$, then we fine-tune the pre-trained model on the corresponding fine-tuning dataset, i.e., $\left(\mathcal{F}^{\text{DE}}, \mathcal{F}^{\text{AMR}}\right)$.

5.3 Main Results

Table 2 shows the performance of AMR parsing and AMR-to-text generation for German (DE), Spanish (ES), and Italian (IT).

[5] https://www.statmt.org/wmt14/translation-task.html

[6] https://www.statmt.org/europarl/index.html

[7] https://github.com/OpenNMT/OpenNMT-py

Approach	AMR Parsing			AMR-to-Text		
	DE	ES	IT	DE	ES	IT
Baseline$_{\text{scratch}}$	58.10	60.65	58.67	13.11	17.83	13.59
Baseline$_{\text{pre-trained}}$	64.90	68.05	66.54	19.32	27.17	24.13
XLPT-AMR$_{\text{none}}$	48.97	59.52	58.13	10.63	21.17	16.56
XLPT-AMR$_{\text{vanilla}}$	66.88	69.86	69.13	23.11	29.14	27.56
XLPT-AMR$_{\text{one4all}}$	67.40	69.85	69.26	23.37	31.17	28.26
XLPT-AMR$_{\text{targeted}}$	68.31	70.10	69.64	24.15	30.83	28.27
XLPT-AMR$_{\text{T-S}}$	**70.45**	**71.76**	**70.80**	**25.69**	**31.36**	**28.42**
Previous works on cross-lingual AMR parsing						
Damonte and Cohen (2018)†	57.0	60.0	58.0	-	-	-
Blloshmi et al. (2020)‡	53.0	58.0	58.1	-	-	-
Sheth et al. (2021)‡	62.7	67.9	67.4	-	-	-
Previous works on cross-lingual AMR-to-text generation						
Fan and Gardent (2020)‡	-	-	-	15.3	21.7	19.8

Table 2: Performance of AMR parsing in Smatch F1 and AMR-to-text generation in BLEU for German (DE), Spanish (ES), and Italian (IT). Here, XLPT-AMR$_{\text{none}}$ denotes that we test the pre-trained models without fine-tuning them. XLPT-AMR$_{\text{one4all}}$, XLPT-AMR$_{\text{targeted}}$, and XLPT-AMR$_{\text{T-S}}$ indicate that we use one-for-all, targeted and teacher-student as MTL fine-tuning method, respectively. † is for using Google translator while ‡ for pre-trained models.

From the performance comparison of the two baseline approaches, it is not surprising to find out that pre-training on silver datasets is a very effective way to boost performance (Konstas et al., 2017; Xu et al., 2020). By using silver datasets, we obtain improvements of 6.80 ∼ 7.87 Smatch F1, and 6.21 ∼ 10.54 BLEU for parsing and text generation, respectively.

With any of our fine-tuning methods, our cross-lingual pre-training approach further improves the performance over the strong baseline Baseline$_{\text{pre-trained}}$ in both parsing and generation tasks over all languages. It shows that like other fine-tuning methods, vanilla fine-tuning significantly boosts the performance of both parsing and generation. However, it still underperforms any of the MTL fine-tuning methods. This confirms that it is important to optimize for high accuracy of a certain fine-tuning task while preserving the performance of other pre-training. The performance comparison between XLPT-AMR$_{\text{one4all}}$ and XLPT-AMR$_{\text{targeted}}$ suggests that selectively choosing relevant fine-tuning tasks, rather than including all fine-tuning tasks, could further boost parsing and generation performance with the exception of Spanish generation task.

The XLPT-AMR$_{\text{T-S}}$ models perform the best, which reveals that using the teacher-student framework to guide the decoding process also helps the student task. This is owing to fact that the teacher models achieve better performance than the student models. See more in Section 5.4 for performance comparison of teacher and student models.

Finally, we compare our approach to the previous studies. Among them, both Blloshmi et al. (2020) and Fan and Gardent (2020) adopt pre-trained models which cover either the encoder part, or the decoder part. From the results we can see even our baseline Baseline$_{\text{pre-trained}}$ outperforms them by pre-training the encoder and the decoder simultaneously. The results also show that our XLPT-AMR$_{\text{T-S}}$ models greatly advance the state of art. For example, our XLPT-AMR$_{\text{T-S}}$ models outperform Sheth et al. (2021) by 3.4∼7.8 Smatch F1 on AMR parsing of the three languages while surpass Fan and Gardent (2020) by around 10 BLEU on AMR-to-text generation.

Table 3 compares the performance of fine-grained metrics for AMR parsing. It shows that our XLPT-AMR$_{\text{T-S}}$ models achieve the best performance on all the metrics with the only exception of Concepts for Italian AMR parsing. It shows that like English AMR parsing, all models predict Reentrancies poorly (Szubert et al., 2020). It also demonstrates that Negations is another metric which is hard to predict. In future work, we will pay particular attention to the two metrics.

Metric	Blloshmi et al. (2020)			$\textbf{Baseline}_{\textbf{pre-trained}}$			$\textbf{XLPT-AMR}_{\textbf{T-S}}$		
	DE	**ES**	**IT**	**DE**	**ES**	**IT**	**DE**	**ES**	**IT**
Smatch	53.0	58.0	58.1	64.90	68.05	66.54	**70.45**	**71.76**	**70.80**
Unlabeled	57.7	63.0	63.4	69.53	72.49	71.16	**74.57**	**75.86**	**75.07**
No WSD	53.2	58.4	58.4	65.16	68.40	66.78	**70.70**	**72.14**	**71.11**
Concepts	58.0	65.9	64.7	68.79	73.06	**78.21**	**73.42**	**76.29**	74.86
Named Ent.	66.0	65.9	64.7	79.12	81.34	68.42	**85.95**	**84.09**	**83.35**
Negations	11.7	23.4	29.2	42.69	51.93	48.57	**52.48**	**57.19**	**54.95**
Wikification	60.9	63.1	67.0	67.40	69.40	71.05	**74.05**	**73.32**	**73.73**
Reentrancies	39.9	46.6	46.1	42.40	46.20	44.10	**45.70**	**48.40**	**47.90**
SRL	47.9	55.2	54.7	60.50	65.20	63.80	**64.90**	**68.50**	**67.30**

Table 3: Fine-grained F1 scores of AMR parsing.

5.4 Discussion

In this section, we try to answer the following three questions:

- First, what is the performance of teacher models when we use teacher models to guide student ones in teacher-student-based MTL fine-tuning?
- Second, what is the effect of the two machine translation tasks in pre-training?
- Third, in our approach we take English as pivot language by taking advantage of large scale English-to-German (or Spanish, Italian) dataset. What is the performance of English AMR parsing and AMT-to-text generation?

Performance of teacher models in teacher-student-based MTL fine-tuning. Table 4 compares the performance of teacher and student models. It shows that the performance of teacher models for English AMR parsing and English-to-X translation is much higher than the counterparts of student models (i.e., Stu.$_{\text{(before)}}$ in the table). The table also shows that the student models beneift from receiving guidance from the teachers. For example, while the English AMR parsing model (i.e., the teacher) achieves 78.62 Smatch F1 on the test set, it improves the performance of the German AMR parsing model (i.e., the student) from 68.31 Smatch F1 to 70.45. Similarly, while the English-to-German model (i.e., the teacher) achieves 39.40 BLEU on the test set, it boosts the performance of the German AMR-to-text generation model (i.e., the student) from 24.15 BLEU to 25.69.

Effect of machine translation tasks in pre-training. We use German as a representative. Note that when machine translation tasks are not involved in pre-training, the targeted MTL fine-tuning method is not applicable since we cannot use machine translation as the auxiliary task. Therefore, we use the vanilla fine-tuning method to fine-tune the pre-trained models. Table 5 compares the performance with/without machine translation tasks in pre-training. From it, we observe that including machine translation tasks in pre-training achieves improvements of 2.77 Smatch F1 and 2.46 BLEU on German AMR parsing and text generation, respectively. This suggests the necessity to have machine translation tasks in pre-training.

Performance of English AMR parsing and AMR-to-Text generation. Based on the pre-trained models, we take the targeted MTL fine-tuning method (Section 4.3) as a representative. Specifically, for English AMR parsing, we choose English-to-X $(X \in \{\text{German}, \text{Spanish}, \text{Italian}\})$ as the auxiliary fine-tuning task while for English test generation, we choose X-to-English as the auxiliary task.

Table 6 shows that the performance of English parsing and generation is much higher than that of other languages. Moreover, we find that the results of English AMR parsing are quite close when combining English with any of other languages whereas the results of English AMR-to-text generation are considerably different. One possible reason for the phenomenon is that English AMR-to-text generation is relevant to the sizes of machine translation datasets used in pre-training (i.e., 3.9M for EN-DE translation whereas 1.9M for both EN-ES and EN-IT, respectively) while English parsing seems to be less affected by the sizes of (silver) datasets. It indicates that with more English sentences in pre-training, it helps the generation models to generate

Model	AMR Parsing			AMR-to-Text		
	DE	ES	IT	DE	ES	IT
Teacher	78.62	78.16	78.58	39.40	40.41	36.67
Stu.(before)	68.31	70.10	69.64	24.15	30.83	28.27
Stu.(after)	70.45	71.76	70.80	25.69	31.36	28.42

Table 4: Performance comparison of teacher and student models. Note that the performance of teacher models is for English AMR parsing, and English-to-X translation, respectively.

Pre-training tasks	AMR Parsing	AMR-to-Text
All	66.88	23.11
- MT tasks	64.11	20.65

Table 5: Performance comparison for German with/without machine translation tasks in pre-training.

Language	AMR Parsing	AMR-to-Text
DE	68.31	24.15
EN	78.62	40.89
ES	70.10	30.83
EN	78.16	32.29
IT	69.64	28.27
EN	78.58	31.98

Table 6: Performance comparison for AMR parsing and AMR-to-text generation for English and other three zero-shot languages.

more fluent and correct English sentences.

6 Conclusions

In this paper we proposed a cross-lingual pre-training approach via multi-task learning for zero-shot AMR parsing and AMR-to-text generation. Upon English AMR dataset and English-to-X parallel datasets, we pre-trained models on three types of relevant tasks, including AMR parsing, AMR-to-text generation, and machine translation. We also explored and compared four different fine-tuning methods. Experimentation on the multilingual AMR dataset shows that our approach greatly advances the state of the art.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant No. 2020AAA0108600 and by the National Natural Science Foundation of China under Grant No. 61876120.

References

Rafael Anchiêta and Thiago Pardo. 2020. Semantically inspired amr alignment for the Portuguese language. In *Proceedings of EMNLP*, pages 1595–1600.

Xuefeng Bai, Linfeng Song, and Yue Zhang. 2020. Online back-parsing for AMR-to-text generation. In *Proceedings of EMNLP*, pages 1206–1219.

Miguel Ballesteros and Yaser Al-Onaizan. 2017. Amr parsing using stack-lstms. In *Proceedings of EMNLP*, pages 1269–1275.

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.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of ACL*, pages 273–283.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of AAAI*.

Rexhina Blloshmi, Rocco Tripodi, and Roberto Navigli. 2020. XL-AMR: Enabling cross-lingual AMR parsing with transfer learning techniques. In *Proceedings of EMNLP*, pages 2487–2500.

Deng Cai and Wai Lam. 2019. Core semantic first: A top-down approach for AMR parsing. In *Proceedings of EMNLP*, pages 3799–3809.

Deng Cai and Wai Lam. 2020a. AMR parsing via graph⇌sequence iterative inference. In *Proceedings of ACL*, pages 1290–1301.

Deng Cai and Wai Lam. 2020b. Graph transformer for graph-to-sequence learning. In *Proceedings of AAAI*, pages 7464–7471.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structure. In *Proceedings of ACL*, pages 748–752.

Kris Cao and Stephen Clark. 2019. Factorising AMR generation through syntax. In *Proceedings of NAACL*, pages 2157–2163.

Yun Chen, Yang Liu, Yong Cheng, and Victor O.K. Li. 2017. A teacher-student framework for zero-resource neural machine translation. In *Proceedings of ACL*, pages 1925–1935.

Marco Damonte and Shay B. Cohen. 2018. Cross-lingual abstract meaning representation parsing. In *Proceedings of NAACL*, pages 1146–1155.

Marco Damonte and Shay B. Cohen. 2019. Structural neural encoders for AMR-to-text generation. In *Proceedings of NAACL*, pages 3649–3658.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for abstract meaning representation. In *Proceedings of EACL*, pages 536–546.

Angela Fan and Claire Gardent. 2020. Multilingual AMR-to-text generation. In *Proceedings of EMNLP*, pages 2889–2901.

Thiago Castro Ferreira, Iacer Calixto, Sander Wubben, and Emiel Krahmer. 2017. Linguistic realisation as machine translation: Comparing different mt models for amr-to-text generation. In *Proceedings of INLG*, pages 1–10.

Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime Carbonell. 2016. Generation from abstract meaning representation using tree transducers. In *Proceedings of NAACL*, pages 731–739.

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 ACL*, pages 1426–1436.

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

Jonas Groschwitz, Matthias Lindemann, Meaghan Fowlie, Mark Johnson, and Alexander Koller. 2018. AMR dependency parsing with a typed semantic algebra. In *Proceedings of ACL*, pages 1831–1841.

Zhijiang Guo and Wei Lu. 2018. Better transition-based AMR parsing with a refined search space. In *Proceedings of EMNLP*, pages 1712–1722.

Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. 2019. Densely connected graph convolutional networks for graph-to-sequence learning. *TACL*, 7:297–312.

Jan Hajič, Ondšová Bojar, and Zdeňka Urešová. 2014. Comparing Czech and English AMRs. In *Proceedings of Workshop on Lexical and Grammatical Resources for Language Processing*, pages 55–64.

Hamza Harkous, Isabel Groves, and Amir Saffari. 2020. Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity. In *Proceedings of COLING*, pages 2410–2424.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of EMNLP*, pages 1317–1327.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL, System Demonstrations*, pages 67–72.

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 ACL*, pages 146–157.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Proceedings of NeurIPS*.

Zhizhong Li and Derek Hoiem. 2018. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947.

Chunchuan Lyu and Ivan Titov. 2018. AMR parsing as graph prediction with latent alignment. In *Proceedings of ACL*, pages 397–407.

Manuel Mager, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos. 2020. GPT-too: A language-model-first approach for AMR-to-text generation. In *Proceedings of ACL*, pages 1846–1852.

Rik van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representation. *Computational Linguistics in the Netherlands Journal*, 7:93–108.

Kishore Papineni, Salim Roukos, Ward Todd, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Xiaochang Peng, Chuang Wang, Daniel Gildea, and Nianwen Xue. 2017. Addressing the data sparsity issue in neural AMR parsing. In *Proceedings of EACL*, pages 366–375.

Nima Pourdamghani, Kevin Knight, and Ulf Hermjakob. 2016. Generating English from abstract meaning representations. In *Proceedings of INLG*, pages 21–25.

Leonardo F. R. Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing AMR-to-text generation with dual graph representations. In *Proceedings of EMNLP-IJCNLP*, pages 3183–3194.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating pretrained language models for graph-to-text generation. In *Computing Research Repository, arXiv:2007.08426*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of ACL*, pages 1715–1725.

Janaki Sheth, Young-Suk Lee, Ramón Fernandez Astudillo, Tahira Naseem, Radu Florian, Salim Roukos, and Todd Ward. 2021. Bootstrapping multilingual amr with contextual word alignments. In *Proceedings of EACL*, pages 394–404.

Marco Antonio Sobrevilla Cabezudo, Simon Mille, and Thiago Pardo. 2019. Back-translation as strategy to tackle the lack of corpus in natural language generation from semantic representations. In *Proceedings of MSR*, pages 94–103.

Linfeng Song, Xiaochang Peng, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2017. AMR-to-text generation with synchronous node replacement grammar. In *Proceedings of ACL*, pages 7–13.

Linfeng Song, Ante Wang, Jinsong Su, Yue Zhang, Kun Xu, Yubin Ge, and Dong Yu. 2020. Structural information preserving for graph-to-text generation. In *Proceedings of ACL*, pages 7987–7998.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of ACL*, pages 1616–1626.

Ida Szubert, Marco Damonte, Shay B. Cohen, and Mark Steedman. 2020. The role of reentrancies in abstract meaning representation parsing. In *Findings of EMNLP*, pages 2198–2207.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N.Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*, pages 5998–6008.

Chuan Wang, Bin Li, and Nianwen Xue. 2018. Transition-based Chinese amr parsing. In *Proceedings of NAACL*, pages 247–252.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015a. Boosting transition-based AMR parsing with refined actions and auxiliary analyzers. In *Proceedings of ACL*, pages 857–862.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015b. A transition-based algorithm for AMR parsing. In *Proceedings of NAACL*, pages 366–375.

Keenon Werling, Gabor Angeli, and Christoerpher D. Manning. 2015. Robust subgraph generation improves abstract meaning representation parsing. In *Proceedings of ACL*, pages 982–991.

Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2020. Improving AMR parsing with sequence-to-sequence pre-training. In *Proceedings of EMNLP*, pages 2501–2511.

Nianwen Xue, Ond$\check{s}$ov$\acute{a}$ Bojar, Jan Haji$\check{c}$, Martha Palmer, Zde$\breve{n}$ka Ure$\check{s}$ov$\acute{a}$, and Xiuhong Zhang. 2014. Not an interlingua, but close: Comparison of English AMRs to Chinese and Czech. In *Proceedings of LREC*, pages 1765–1772.

Shaowei Yao, Tianming Wang, and Xiaojun Wan. 2020. Heterogeneous graph transformer for graph-to-sequence learning. In *Proceedings of ACL*, pages 7145–7154.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019a. AMR parsing as sequence-to-graph transduction. In *Proceedings of ACL*, pages 80–94.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019b. Broad-coverage semantic parsing as transduction. In *Proceedings of EMNLP-IJCNLP*, pages 3786–3798.

Yanbin Zhao, Lu Chen, Zhi Chen, Ruisheng Cao, Su Zhu, and Kai Yu. 2020. Line graph enhanced AMR-to-text generation with mix-order graph attention networks. In *Proceedings of ACL*, pages 732–741.

Jiawei Zhou, Tahira Naseem, Ramón Fernandez Astudillo, and Radu Florian. 2021. Amr parsing with action-pointer transformer. In *Proceedings of NAACL*, pages 5585–5598.

Junsheng Zhou, Feiyu Xu, Hans Uszkoreit, Weiguang Qu, Ran Li, and Yanhui Gu. 2016. AMR parsing with an incremental joint model. In *Proceedings of EMNLP*, pages 680–689.

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 EMNLP-IJCNLP*, pages 5459–5468.

Task	BLEU
EN-DE	28.67
EN-ES	26.54
EN-IT	26.79

Table 7: Performance in BLEU score for the three translation tasks.

A English AMR Parser on AMR 2.0

Our English AMR parser is learned in a seq2seq framework and trained on AMR2.0, which consists of 36,521 training AMRs, 1,368 development AMRs and 1,371 testing AMRs. We share vocabulary for the input and the output by segmenting tokens into pieces by byte pair encoding (BPE) with 20K merge operations.

We use *OpenNMT-py* as the implementation of Transformer. In model setting, we use Transformer base model setting. We use Adam with $\beta_1 = 0.9$, $\beta_2 = 0.98$ for optimization. Batch size, learning rate, warm-up step, and dropout rate are set to 4096, 2.0, 16000 and 0.1 respectively. We train the model for 250K steps on 1 GPUs and save models every 10K steps. Finally, we obtain final model by averaging the last 10 checkpoints.

The English AMR parser achieves 73.68 and 73.24 Smatch F1 on the dev and test set, respectively.

B NMT Models for English-to-German, English-to-Spanish, English-to-Italian

In pre-processing, we tokenize all of MT corpus with Moses scripts.[8] Then we segment words into pieces by BPE with 32K (30K) BPE merge operations for EN-DE (both EN-ES and EN-IT). After filtering long and imbalanced pairs, we get 3.9M parallel sentence pairs for EN-DE and 1.9M for both EN-ES and EN-IT.

We again use *OpenNMT-py* as the implementation of Transformer. In model setting, we use Transformer big model setting. We use Adam with $\beta_1 = 0.9$, $\beta_2 = 0.998$ for optimization. Batch size, learning rate, warm-up step, and dropout rate are set to 8192, 2.0, 8000 (16000 for both EN-ES and EN-IT) and 0.1, respectively. We train the model for 100K (110K for EN-ES and 150K for EN-IT) steps on 4 GPUs and save models very 5000 steps. For each translation task, we obtain final model by averaging the last 5 (20 for both EN-ES and EN-IT) checkpoints.

For evaluation, we use case-sensitive BLEU measured by multi-bleu script. Table 7 shows the performance of the three translation models on the test sets, i.e., newstest2014 for EN-DE and newstest2009 for both EN-ES and EN-IT.

[8] https://github.com/moses-smt/mosesdecoder