Should Cross-Lingual AMR Parsing go Meta? An Empirical Assessment of Meta-Learning and Joint Learning AMR Parsing

Anonymous ACL submission

Abstract

Cross-lingual AMR parsing is the task of predicting AMR graphs in a target language when training data is available only in a source language. Due to the small size of AMR training data and evaluation data, cross-lingual AMR parsing has only been explored in a small set of languages such as English, Spanish, German, Chinese, and Italian. Taking inspiration from Langedijk et al. (2022), who apply meta-learning to tackle cross-lingual syntactic parsing, we investigate the use of meta-learning for cross-lingual AMR parsing. We evaluate our models in both zero-shot and few-shot scenarios and assess their effectiveness in Croatian, Farsi, Korean, Chinese, and French. Notably, Korean and Croatian test sets are developed as part of our work, based on the existing *The Little Prince* English AMR corpus, and made publicly available. We empirically study this approach by comparing it to a classical joint learning method. Our findings suggest that while the meta-learning model performs similarly to a jointly trained model on average SMATCH score, it exhibits inconsistency and unstable performance across settings.

1 Introduction

Abstract Meaning Representation (Banarescu et al., 2013, AMR) represents the meaning of texts as rooted and directed acyclic graphs. AMR graphs capture the underlying semantics of input texts while abstracting away from their syntactic realizations. Nodes in AMR graphs are not explicitly mapped to their input token. Hence, it is an unanchored formalism. AMRs are widely used to enhance the capabilities of NLP systems such as question answering (Deng et al., 2022; Kapanipathi et al., 2021), text summarization (Liao et al., 2018; Liu et al., 2015), or human-robot interaction (Bonial et al., 2019, 2023).

AMR was originally designed for English texts only. However, Damonte and Cohen (2018) demonstrated that AMR could be used for other languages such as Spanish, Italian, Chinese, and German. Since then, many approaches have adopted AMR parsing for multilingual AMR parsing (Procopio et al., 2021; Blloshmi et al., 2020; Xu et al., 2021; Cai et al., 2021; Sheth et al., 2021). However, one of the main challenges for this task is the lack of data. Currently, training data are only available in English (Knight et al., 2017, 2020) and evaluation data in 6 languages: English, German, Spanish, Italian, Chinese (Damonte and Cohen, 2018), and French (Kang et al., 2023). To overcome the lack of training data in target languages, previous approaches create silver training data in the target languages through translation (Damonte and Cohen, 2018; Blloshmi et al., 2020), or using parallel corpus with English AMR parsers (Xu et al., 2021; Blloshmi et al., 2020). Another approach uses English data for training and then evaluates the model in the target language as a zero-shot approach (Procopio et al., 2021). Since evaluation data is available in five languages, most of these proposals focus on this small set of languages.

In this study, our goal is to apply AMR parsing for more diverse languages that have been less explored in previous approaches and tackle the lack of training data with few-shot learning. Taking inspiration from Langedijk et al. (2022), who applied meta-learning for few-shot cross-lingual syntactic parsing, we apply meta-learning for cross-lingual AMR parsing. To examine the efficiency of the method, we compare the meta-learning approach to a classical joint learning method. We focus on specific settings such as the number of training languages, the robustness of the model with respect to input translation quality, low-resource settings, and various hyperparameters for fine-tuning.

Our contributions to cross-lingual AMR parsing are threefold:

- This work presents the **first empirical study on meta-learning applications on cross-lingual AMR parsing**.

- We train and evaluate our model in languages less explored for AMR parsing: Korean, Croatian, French, and Farsi.
- We **publish new evaluation data in Korean and Croatian**, based on *The Little Prince*.
- We release a multilingual AMR parser that can be evaluated in many languages in zero-shot. We also release the code to train and evaluate the model.

2 Related Work

2.1 Cross-Lingual AMR Parsing

Cross-lingual AMR parsing tasks refer to predicting AMR graphs in a target language when training data is available only in a source language (target language $\neq$ source language). AMR training data, consisting of pairs made of a sentence[1] and its corresponding AMR graph, are only available in English. Therefore, previous approaches have either created artificial training data in the target language or trained the model using English AMR data, subsequently evaluating it in the target language in a zero-shot fashion.

Damonte and Cohen (2018) adopt machine translation to translate English sentences in the training data into the target language to obtain training data in target languages. The translations are *silver* due to their quality as opposed to *gold*, which is manually annotated. They also adopt annotation projection with word alignment to obtain training data in target languages. Xu et al. (2021) and Blloshmi et al. (2020) adopt parallel corpus (English - target language) and parse the English side of the corpus with an existing English AMR parser to eventually obtain a new pair of target text and its corresponding AMR graph (the data is *silver* due to the quality of the AMR graph). Conversely, in the zero-shot approach, the English AMR task is considered a pivot task, and multilingual translation between English and the target language is added as an auxiliary task (Procopio et al., 2021; Xu et al., 2021). The second task allows a model to parse AMR graphs from the target language in zero-shot. Uhrig et al. (2021) propose to translate target texts into English and then use an existing English parser to obtain its graph. This simple method does not require training an AMR parser in the target languages and provides a simple yet effective solution for cross-lingual AMR parsing.

[1]AMR graph can be used beyond sentence level (O'Gorman et al., 2018).

However, these approaches focus on a small set of languages for which training or evaluation data are available. We extend our research to a more diverse set of languages. To obtain training data in different languages, we employ machine translation as in Damonte and Cohen (2018) and use the data to train a multilingual AMR parser. We then evaluate our model in a zero-shot / few-shot fashion on five languages: Chinese (Sino-Tibetan), Korean (Koreanic), and three languages from three branches of the Indo-European family: French (Romance), Farsi (Indo-Iranian) and Croatian (Slavic).

2.2 Meta Learning

Meta-learning, also known as *learning to learn*, is a learning paradigm that allows a model to quickly learn a new task with only a few examples. This is made possible by the prior knowledge that the model has acquired through a series of different tasks. There are three main approaches to meta-learning: *metric-based* meta-learning, *model-based* meta-learning, and *optimization-based* meta-learning. Among them, the optimization-based method is widely used in NLP applications due to its effectiveness. Especially, model agnostic meta-learning (Finn et al., 2017, MAML) under this category has gained popularity due to its efficacy. Previous approaches have adopted MAML in few-shot scenarios for question answering (Nooralahzadeh et al., 2020), machine translation (Gu et al., 2018), speech recognition (Singh et al., 2022), dependency parsing (Langedijk et al., 2022) among others.

The idea behind MAML is to find good initial parameters θ that can be tuned to unseen tasks with only a few optimization steps and a few training data examples. MAML trains a model to be good at adapting to new tasks only with a few examples by *simulating the few-shot training and evaluation* during the training. At each iteration step, a base model is temporarily trained with a few examples and then evaluated to unseen examples of the task. The loss calculated in this step contains information about how good the model is at predicting unseen examples after being trained on a few examples. The global learning objective is to minimize this loss. Therefore, over the entire training, the model learns to adapt quickly using only a few examples. Moreover, the model is trained with different tasks so that it can learn to adapt quickly to any similar tasks.[2] In cross-lingual applications, each task cor-

[2]Target tasks with a similar distribution as the source tasks

responds to a different language, which is the focus of our study.

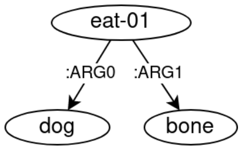
The closest approach to ours is Langedijk et al. (2022), who adopt MAML for cross-lingual dependency parsing. They train a dependency parser on a set of languages using MAML and then evaluate their model on unseen languages to investigate the model's ability to adapt quickly. In contrast, we focus on a *semantic* parsing task with an unanchored formalism. In addition, they have multilingual training data at hand, whereas we generate our silver multilingual data by machine translation from English data. Another difference is that they use a graph-based biaffine model for parsing, whereas we use a seq2seq model with a linearized graph. Sherborne and Lapata (2023) applied meta-learning to cross-lingual SQL parsing. While useful at representing (and executing) database queries expressed in natural language, SQL is not a general-purpose semantic formalism like AMR. To the best of our knowledge, our work is the first to apply MAML for cross-lingual AMR parsing.

3 Meta X-AMR

3.1 Seq2seq AMR Parsing

Three AMR parsing approaches are widely used: transition-based parsing (Damonte et al., 2017), graph-based parsing (Zhang et al., 2019; Cai and Lam, 2019), and sequence-to-sequence parsing (Bevilacqua et al., 2021). Among these, the last approach views AMR parsing as generating an AMR graph from input texts using a sequence-to-sequence model. In this approach, AMR graphs should be first linearized in a single-line format (see Figure 1). Bevilacqua et al. (2021) explore various methods to linearize AMR graphs such as depth-first search, breadth-first search, and PENMAN notation. Among these techniques, we adopt depth-first search, identified as the most efficient way for seq2seq AMR parsing according to Bevilacqua et al. (2021). In particular, we use the method and implementation of van Noord and Bos (2017)[3] to linearize AMR graphs. This method includes light pre-processing such as removing variables and wiki links. We refer the readers to van Noord and Bos (2017) for a comprehensive understanding of the linearization process.

To generate AMR graphs from multi-lingual inputs, we employ `mBart-large-50` model (Tang et al., 2020)[4] as done by Procopio et al. (2021). The mBart model is a pretrained transformer (Vaswani et al., 2017) and contains multiple layers of encoders and decoders. For input and output sequences, mBart takes a special token which indicates a language type at the beginning of texts. For multilingual input sentences, we add the prefix according to its language and for AMR graphs, we add an <amr> prefix. Since this is not included in the vocabulary of mBart, we add this new vocabulary to the model and randomly initialize the corresponding vector as done by Procopio et al. (2021). Since the output of this model is a linearized graph, we restructure the AMR graph through post-processing steps for evaluation. We employ the implementation code of van Noord and Bos (2017) for this step.



(a) AMR graph

```
(e / eat-01:ARG0
(d / dog) :ARG1
(b / bone))
```

(b) linearized AMR

Figure 1: "The dog eats a bone."

3.2 MAML for Cross-lingual AMR Parsing

We apply MAML (Finn et al., 2017) to train our multilingual AMR parser. The training procedure is described below (see Figure 2 for visual description).

Step 1: At each iteration step, the initial model (Θ) is copied once per language i. For each i, $2 \times K$ examples are randomly sampled from D_i^{train} and divided into the support and the query set (K each). Using the support set, the model is temporarily updated with stochastic gradient descent with learning rate α (Eq. 1). Iterate through the support set for P adaptation steps to obtain Φ_i:

$$\Phi_i \leftarrow \Theta - \alpha \bigtriangledown_\Theta \mathcal{L}(\Theta_i). \quad (1)$$

Next, the loss is computed to evaluate the temporary model Φ_i on the query set. The loss $\mathcal{L}_i(\Phi_i)$ is saved for the next step. The entire step is called an 'inner loop' and the inner loop is repeated over the entire task batch, that is, for the number of all training languages I.

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

[4] We use `facebook/mbart-large-50` model via the transformers library (Wolf et al., 2020).

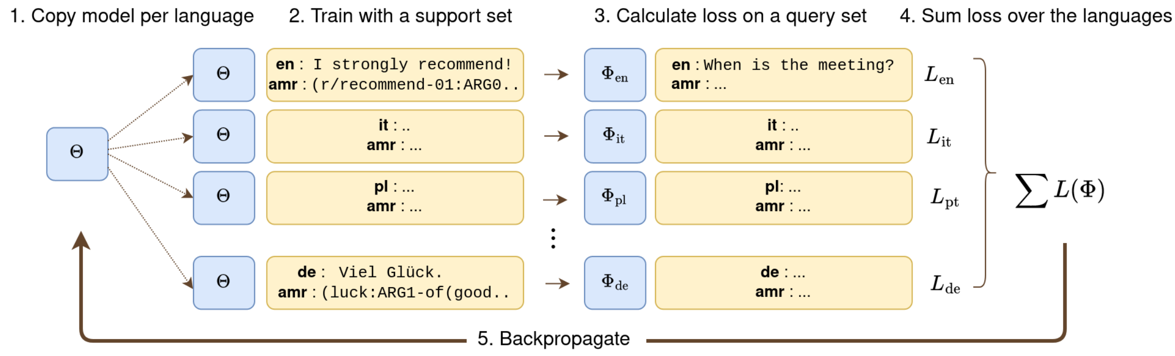


Figure 2: One training step for MAML cross-lingual AMR parsing.

Step 2: $\mathcal{L}_i(\Phi_i)$ is summed up over training languages to update the initial model Θ by stochastic gradient descent with a learning rate β. This entire step is called an 'outer loop'. Note that in Eq. 2, we use $\nabla_{\Phi_i}\mathcal{L}_i(\Phi_i)$ instead of $\nabla_{\theta}\mathcal{L}_i(\Phi_i)$ because we apply First-Order MAML to avoid computation overhead (second-order derivative requires heavy computation):

$$\Theta \leftarrow \Theta - \beta \sum_i \nabla_{\Phi_i}\mathcal{L}_i(\Phi_i). \qquad (2)$$

Step 3: Repeat Step 1 and Step 2 until the total number of training steps.
Step 4: We evaluate the model with target test data once the training is over. The evaluation is done in a zero-shot or a few-shot fashion, which means that the model is evaluated on new target languages that are unseen during the training.

4 Experimental Setup

4.1 Data

We aim to train a multilingual AMR parser that adapts quickly to new languages, specifically French, Chinese, Korean, Farsi, and Croatian, with 0 example (0-shot learning) or a few examples (few-shot learning). Our method is similar to that of Langedijk et al. (2022) in applying meta-learning for a few-shot cross-lingual parsing task, but our training data is only available in English, whereas they have multilingual training data. To create multilingual training data, we apply machine translation as in previous approaches (Damonte and Cohen, 2018; Xu et al., 2021; Blloshmi et al., 2020). We adopt DeepL[5] for automatic translation and translate English AMR training data (LDC2020T02) (Knight et al., 2020) into 13 languages: German, Italian, Romanian, Finnish, Russian, Turkish, Japanese, Czech, Dutch, Polish, Swedish, Estonian, and Indonesian. These languages are supported by mBart (Tang et al., 2020), a model we adopted for our experiments. The 13 languages were chosen for language diversity and cover 5 language families: Indo-European (Germanic, Romance, Slavic), Uralic, Turkic, Japonic, and Austronesian. For each training language, there are 55,635 pairs of a sentence and their corresponding AMR graph. We use a total of 14 languages including English for our training data. We use Spanish as the validation language. Note that we need the validation set as well as k fine-tuning examples in the same language for k-shot evaluation. For the validation set, we use the Spanish test set from AMR 2.0 (Damonte and Cohen, 2020) and for the fine-tuning dataset, we translate k random examples of the English dev set. We use French, Chinese, Korean, Farsi, and Croatian as test languages. For French, Chinese, and Farsi, we employ the Little Prince AMR corpus annotated in each language, respectively from Kang et al. (2023), `https://amr.isi.edu/` and Takhshid et al. (2022)[6]. For Croatian and Korean, we create our test sets by manually aligning The Little Prince corpus in each language to corresponding English AMR graphs. We make the test set publicly available.[7]

4.2 Meta-Training and Evaluation

We adopt mBart model from the transformers library (Wolf et al., 2020) to train our multilingual AMR parser. To implement model-agnostic meta-learning, we employ the `learn2learn` library

[5] `https://www.deepl.com`

[6] The original dataset in Farsi consists of AMR graphs where the nodes are in Farsi. Since we employ AMR graphs with English nodes, we use only the input texts of the corpus and use graphs from the English AMR corpus.

[7] The URL will be provided upon publication.

(Arnold et al., 2020). We train our model for 30,000 steps and evaluate the model every 500 steps with the Spanish validation set. Early stopping is applied, terminating training if the dev SMATCH score fails to improve for more than 7,500 steps. For both validation and testing, we employ k-shot learning, where the model is fine-tuned with k examples before being evaluated on the entire test/validation set. The number of fine-tuning cycles, called an adaptation step, is denoted as P. Unless specified otherwise, we set $P = 0$ and $k = 0$ (0-shot learning). MAML requires two learning rates, one for the inner loop (α) and one for the outer loop (β). We conducted a grid search to identify an optimal learning rate set and used $\alpha = 1 \times 10^{-5}$, $\beta = 3 \times 10^{-5}$ throughout the experiments. For β, we use a linear learning rate scheduler with 1,500 warm-up steps. Unless specified otherwise, we apply 1×10^{-5} to fine-tune a model before validation/testing. At each iteration step during the training, $2 \times K$ are sampled to form a query and a support set for each training language. As a result, the batch size N equals $2 \times K \times I$, where I denotes the number of training languages. By default, we assign $K = 8$ and $I = 14$, unless stated otherwise. We report evaluation scores using SMATCH (Cai and Knight, 2013), an evaluation metric for AMR graphs.

4.3 Baseline with Joint Learning

We train a baseline model with a joint learning method for comparison with our approach. The same mBart model is used as described in 4.2. For the training set, we use the multilingual AMR training sets in 14 languages described in 4.1. At each iteration step, we randomly select N training examples from the concatenated training sets to calculate the loss and optimize the model accordingly. The model is evaluated in 0-shot or k-shot depending on the experiment setting (details are described for in each Section 5). Note that we aim to conduct a comparative study with the meta-learning approach. Therefore, unless mentioned otherwise, we apply the same hyperparameters and test/evaluation method for both approaches (e.g. batch size, learning rate scheduler, k-shot size). However, whereas meta-learning requires two learning rates for an inner loop and an outer loop, the baseline only requires one learning rate during the training. We use a uniform learning rate for training 3×10^{-5} with a linear scheduler with 1500 warm-up steps.

	fr	zh	ko	fa	hr	avg
base_14langs	56.3	45.6	42.1	46.3	51.4	48.3
base_12langs	53.6	41.6	40.1	43.4	45.9	44.9
base_8langs	47.5	39.8	39.1	40.5	22.4	37.8
MAML_14langs	56.5	46.1	42.2	46.7	50.8	48.4
MAML_12langs	48.5	39.4	35.1	39.7	45.0	41.5
MAML_8langs	47.7	39.6	34.3	40.1	42.4	40.8

Table 1: SMATCH scores according to the number of training languages.

5 Research Questions and Discussions

We examine the strengths and weaknesses of our method by answering the research questions below. For evaluation, we systematically vary six hyperparameters individually while keeping the remaining parameters fixed, and assess their influence on the model's performance through comprehensive evaluation and analysis (see Table 8 of Appendix A for the entire hyper-parmeters settings). We compare each model to its opponent baseline model for evaluation. Questions Q1 to Q3 center on how the two models respond to specific factors during the training phase, while Q4 to Q6 pertain to the fine-tuning and evaluation stages. The discussions on the questions lead to a final discussion Q7 on whether meta-learning proves to be the optimal approach for cross-lingual AMR parsing.

Q1: How does the number of languages affect the performance of the models?

To examine how the number of training languages impacts the model performance, we incrementally add more languages to the training data and we train three models respectively with 8, 12, and 14 languages. The first model is trained in German, English, Italian, Romanian, Russian, Turkish, and Japanese. Then we add Czech, Dutch, Polish, and Swedish, and then finally we add Estonian and Indonesian. Note that for meta-learning, the batch size depends on the number of training tasks since we randomly sample K examples per language (batch size $= 2 \times K \times I$ where I denotes the number of training languages). To keep the batch size consistent across experiments while altering only the number of languages, when more than 8 languages are used for training, we randomly sample 8 languages per iteration step and select K training examples per language. Unless specified otherwise, each model is evaluated in a zero-shot manner for five languages: French, Chinese, Korean, Farsi, and Croatian.

Results Table 1 shows that both the MAML and baseline models have a positive correlation with the number of training languages. The baseline model has the largest gain when increasing the number of languages from 8 to 12 language by 15.7%. MAML models, on the other hand, have the biggest gain when increasing the number of languages from 12 to 14 languages by 14.2%. Looking in detail per target language, however, in the MAML model, not all target languages benefit from adding more training languages. Comparing the two MAML models, trained respectively with 8 languages and 12 languages, the SMATCH score drops in Chinese and Farsi when adding four languages to the training data, whereas the baseline model shows a steady increase across target languages when adding more languages. In other words, the baseline model benefits uniformly from the inclusion of more training languages across all target languages, while the performance of the MAML model varies depending on the specific target language. In the MAML models, certain languages experience a decrease in performance despite the addition of more training languages. A caveat of this experiment is that the results may depend on the order in which the languages are added and their typological relationship to evaluation languages (we leave this investigation to future work).

Q2: How robust is the model with respect to translation quality?

To assess the impact of the translation source on our method, we employ an alternative translation model to translate our training data. Specifically, we use the mBart translation models, sourced from the Huggingface hub[8], to translate our training data into 14 languages. Subsequently, we use this translated data to train both the MAML and baseline models. Following this, we contrast the evaluation outcomes of these models with those trained using the DeepL translation.

Results For both the MAML and the baseline models, when using an open-source translation model mBart, the performance drops (see Table 2). In both cases, the Korean SMATCH score drops the most when using the mBart translation model. MAML model is more affected by this change. On the average score, the baseline model drops by 0.9%, whereas the MAML-model drops by 2.3%. This result shows that the meta-learning model is more sensitive to the input texts than the baseline model.

[8] https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt

	fr	zh	ko	fa	hr	avg
base_DeepL	56.3	45.6	42.1	46.3	51.4	48.3
base_mBart	56.2	44.5	41.2	46.1	51.3	47.8
MAML_DeepL	56.5	46.1	42.2	46.7	50.8	48.4
MAML_mBart	55.6	45.1	40.8	46.1	48.9	47.3

Table 2: SMATCH scores according to the translation source.

	fr	zh	ko	fa	hr	avg
base_full	56.3	45.6	42.1	46.3	51.4	48.3
base_1000	41.4	35.1	33.3	36.9	38.5	37.0
MAML_full	56.5	46.1	42.2	46.7	50.8	48.4
MAML_1000	38.9	33.9	32.8	36.1	35.0	35.3

Table 3: SMATCH scores according to training data size.

Q3: Does the model learn efficiently in lower-resource settings?

We assess the robustness of our method in low-resource settings where only a smaller fraction of training data is available. To this end, we randomly sample 1,000 examples for each language (the same examples across languages) and use only this sampled data as training data. Since the entire training data is a lot smaller than the original model, we evaluate the model with the dev set every 100 step instead of 500 to save the best model. We also set the max training step to 10,000 instead of 30,000.

Results Table 3 illustrates the SMATCH scores achieved by both the MAML and baseline models under different training conditions: using the entire dataset (base_full, MAML_full) versus using only 1,000 examples (base_1000, MAML_1000). As expected, both models' performance was substantially decreased when trained on a small dataset. Specifically, the MAML model experienced a higher drop in SMATCH score, declining by 27%, compared to the baseline model, which exhibited a decrease of 23.3%. This discrepancy suggests that the MAML model is more susceptible to performance degradation in low-resource scenarios.

Q4: How many adaptation steps does the model need to learn a new task efficiently?

We fine-tune our model with 32 examples (32-shot) on target languages and evaluate the model with the test set in each language. Since the fine-tuning dataset is not available for the target languages, we use DeepL to translate the English dev set to obtain the data. The model is fine-tuned with the entire fine-tuning data iteratively and the number of iterations is called an adaptation step. To assess the influence of the adaptation steps on the model performance, we increase the number and evaluate the model accordingly. To minimize the effect of finetuning data, we sample 32 examples randomly three times and fine-tune and then evaluate the model three times. We then use the average score of the three evaluation processes. The fine-tuning learning rate is fixed to 1×10^{-5} across all experiments.

Results Figure 3 (whose data is also presented in Table 6 of Appendix A) visually represents the average test SMATCH scores across target languages. When the adaptation step is 0, the model is evaluated in zero-shot. Surprisingly, the results indicate that both the MAML and baseline models performed less effectively after fine-tuning. The baseline model exhibits a gradual decline in performance with extended fine-tuning duration, except for a rapid drop and subsequent recovery between 3 and 7 adaptation steps. The MAML model demonstrates an inconsistent pattern, with a substantial drop in performance between 0 and 2 adaptation steps, followed by gradual improvement between 2 and 5 steps before declining again. The results lack a clear, consistent pattern. However, both models perform better when not fine-tuned at all. We propose the hypothesis that the mBart pre-trained model has already enough knowledge of our target languages (French, Chinese, Korean, Farsi, and Croatian), and fine-tuning the model with only a few examples in each language may impair the model's capacity. This could also be attributed to the domain difference between the fine-tuning dataset and the test dataset. The fine-tuning dataset includes content from general fields like news, online forums, journals, and web blogs, whereas the test dataset consists of *The Little Prince*, a novel written in the 1940s. Consequently, the domain shift between the two datasets may have contributed to the model's inability to generalize effectively to the test domain. Another hypothesis is the small size of the fine-tuning dataset, which may have hindered the model's ability to generalize effectively, or an inadequate learning rate leading to undesirable model updates. We delve into the hypotheses on the learning rate and the size of k in subsequent questions.

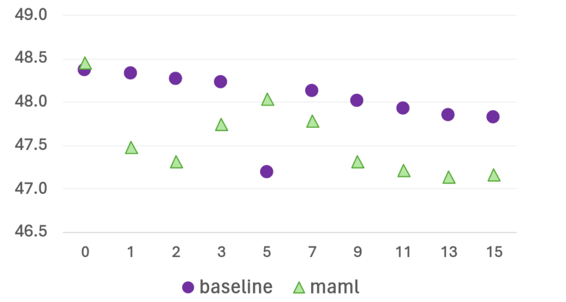


Figure 3: Average SMATCH scores on target languages according to adaptation steps.

Q5: High or low learning rate for fine-tuning?

To examine the model performance depending on different learning rates, we fine-tune our model with different learning rates and then evaluate each model to record test scores. We apply the same settings as in Q4, such as sampling fine-tuning data three times with a k-size equal to 32.

Results Figure 4 (the numerical data is also presented in Table 7 of Appendix A) offers a visual depiction of the mean test SMATCH scores across various target languages. The baseline and the MAML models show a similar pattern that a lower learning rate leads to better results. When the learning rate is 0, that is, the model is not fine-tuned, both models show the best performance. This is aligned with the results in Q4 yet remains questionable as to why fine-tuning in target languages does not lead to a performance gain. This may be caused by small k-size and in the following question, we discuss the results with bigger k-size.

Q6: k-size: the bigger, the better?

To answer this question, we use different $k = 0, 32, 64, 128$ examples for fine-tuning before evaluation. As in Q4 and Q5, the training data is sampled three times and we use the average score. We apply 1×10^{-5} to fine-tune the models.

Results Table 4 illustrates that for values of $32 \leq k \leq 128$, larger values of k result in performance improvements for both models. However, except for the models with 128 fine-tuning

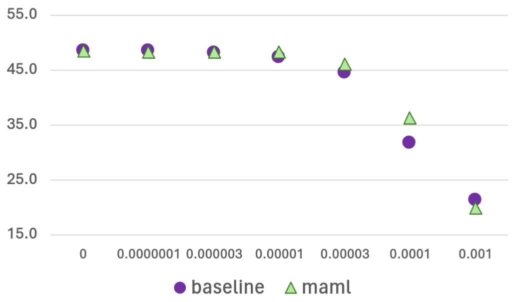


Figure 4: Average SMATCH scores on target languages according to adaptation steps (see Appendix A for numerical results).

k_size	baseline	MAML
0	48.3	48.4
32	48.2	47.3
64	48.2	47.7
128	48.5	48.5

Table 4: Average SMATCH scores on target languages according to k-size.

examples, most models do not exhibit improvement compared to the 0-shot evaluation. It appears paradoxical that a fine-tuned model performs worse than a non-fine-tuned one. Particularly, the MAML model is adversely affected by the fine-tuning step and demonstrates a more significant performance decrease. The most substantial decline is observed between the 0-shot model and the 32-shot model, with a difference of 2.3%, whereas the 32-shot baseline model only degrades by 0.2% compared to the 0-shot model. Consequently, this leads us to revisit the hypotheses discussed in Q4 regarding the prior knowledge of the mBart model in our target languages and the domain shift between the fine-tuning dataset and the test set.

Q7: Should cross-lingual AMR parsing go Meta?

The provided table (Table 5) summarizes the highest SMATCH scores achieved by both the baseline and MAML models during zero-shot evaluation. The difference in performance between these models is marginal, varying depending on the target language. Consequently, drawing a definitive conclusion regarding which method is superior proves challenging. However, through our examination, we have noted that MAML models exhibit greater sensitivity to changes in input types and dataset sizes. Notably, their performance deteriorates significantly in low-resource scenarios or when employing different translation models for inputs. Additionally, inconsistencies arise when fine-tuning the model with varying adaptation steps, complicating result interpretation and impeding progress toward improvement.

	fr	zh	ko	fa	hr	avg
baseline	56.4	45.6	42.1	46.3	51.4	48.36
MAML	56.5	46.1	42.2	46.7	50.8	48.46

Table 5: SMATCH scores of the baseline and the MAML model (0-shot evaluation).

Conversely, our observations indicate that a straightforward joint learning approach yields comparable performance to the MAML model, not only in zero-shot, but also in few-shot evaluations. This highlights that the joint learning method remains a robust starting point for cross-lingual AMR parsing. As a result, MAML does not emerge as the optimal solution for this task, given its inconsistent performance.

6 Conclusion

This study investigates the effectiveness of meta-learning compared to joint learning in cross-lingual AMR parsing. We assess our models across less-explored languages for AMR parsing, including French, Chinese, Korean, Farsi, and Croatian. To facilitate evaluation, we develop new test sets for Korean and Croatian and release the data to promote AMR parsing in diverse languages. We explore various settings to conduct a thorough analysis of the meta-learning approach in contrast to joint learning. Our findings reveal that meta-learning exhibits inconsistent performance across different settings, whereas the joint learning method demonstrates more consistent performance across experimental variations. Consequently, our results suggest that the joint learning method serves as a robust baseline, while meta-learning appears to be a suboptimal approach for cross-lingual AMR parsing. We believe that this research provides valuable insights into the comparative efficacy of meta-learning and joint-learning methods in cross-lingual AMR parsing, offering important guidance for future developments in cross-lingual AMR parsers.

Limitations

Our model does not outperform a simple monolingual model which is trained with AMR data in the target language translated by a MT system. However, our approach can be explored for low-resource languages for which machine translation is not available. In addition, we did not apply grid search to find the best learning rates for the baseline models and used the same learning rate as done by Procopio et al. (2021), who also employed mBart for sequence-to-sequence cross-lingual AMR parsing. This could have affected the results in favor of meta-learning. Nonetheless, this does not affect our conclusion of the empirical study to reveal the weakness of the meta-learning approach for cross-lingual AMR parsing. This study does not include evaluation scores on the AMR 2.0 multilingual test set, which could help position our models relative to the state-of-the-art models. This is because the Spanish test set in AMR 2.0 is already used as our validation set. Therefore, this data is omitted during testing for fair evaluation. Despite the limitations, we believe that our study empirically shows the constraints of meta-learning for cross-lingual AMR parsing and provides valuable insights into the meta-learning application in the task.

References

Sébastien M R Arnold, Praateek Mahajan, Debajyoti Datta, Ian Bunner, and Konstantinos Saitas Zarkias. 2020. learn2learn: A library for Meta-Learning research.

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

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

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

Claire Bonial, Julie Foresta, Nicholas C. Fung, Cory J. Hayes, Philip Osteen, Jacob Arkin, Benned Hedegaard, and Thomas Howard. 2023. Abstract Meaning Representation for grounded human-robot communication. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, pages 34–44, Nancy, France. Association for Computational Linguistics.

Claire N. Bonial, Lucia Donatelli, Jessica Ervin, and Clare R. Voss. 2019. Abstract Meaning Representation for human-robot dialogue. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 236–246.

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, Xin Li, Jackie Chun-Sing Ho, Lidong Bing, and Wai Lam. 2021. Multilingual AMR parsing with noisy knowledge distillation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2778–2789, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Marco Damonte and Shay Cohen. 2020. Abstract meaning representation 2.0 - four translations ldc2020t07.

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

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

Zhenyun Deng, Yonghua Zhu, Yang Chen, Michael Witbrock, and Patricia Riddle. 2022. Interpretable amr-based question decomposition for multi-hop question answering.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Jeongwoo Kang, Maximin Coavoux, Didier Schwab, and Cédric Lopez. 2023. Analyse sémantique AMR pour le français par transfert translingue. In *Actes de CORIA-TALN 2023. Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 2 : travaux de recherche originaux – articles courts*, pages 55–62, Paris, France. ATALA.

Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernandez Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, Dinesh Garg, Alfio Gliozzo, Sairam Gurajada, Hima Karanam, Naweed Khan, Dinesh Khandelwal, Young-Suk Lee, Yunyao Li, Francois Luus, Ndivhuwo Makondo, Nandana Mihindukulasooriya, Tahira Naseem, Sumit Neelam, Lucian Popa, Revanth Gangi Reddy, Ryan Riegel, Gaetano Rossiello, Udit Sharma, G P Shrivatsa Bhargav, and Mo Yu. 2021. Leveraging Abstract Meaning Representation for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3884–3894, Online. Association for Computational Linguistics.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and Nathan Schneider. 2017. Abstract meaning representation (amr) annotation release 2.0 - linguistic data consortium.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and Nathan Schneider. 2020. Abstract meaning representation (amr) annotation release 3.0 - linguistic data consortium.

Anna Langedijk, Verna Dankers, Phillip Lippe, Sander Bos, Bryan Cardenas Guevara, Helen Yannakoudakis, and Ekaterina Shutova. 2022. Meta-learning for fast cross-lingual adaptation in dependency parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8503–8520, Dublin, Ireland. 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.

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.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-shot cross-lingual transfer with meta learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4547–4562, Online. Association for Computational Linguistics.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. AMR beyond the sentence: the multi-sentence AMR corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Luigi Procopio, Rocco Tripodi, and Roberto Navigli. 2021. SGL: Speaking the graph languages of semantic parsing via multilingual translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 325–337, Online. Association for Computational Linguistics.

Tom Sherborne and Mirella Lapata. 2023. Meta-learning a cross-lingual manifold for semantic parsing. *Transactions of the Association for Computational Linguistics*, 11:49–67.

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 the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 394–404, Online. Association for Computational Linguistics.

Satwinder Singh, Ruili Wang, and Feng Hou. 2022. Improved meta learning for low resource speech recognition. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4798–4802.

Reza Takhshid, Razieh Shojaei, Zahra Azin, and Mohammad Bahrani. 2022. Persian abstract meaning representation.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

Sarah Uhrig, Yoalli Garcia, Juri Opitz, and Anette Frank. 2021. Translate, then parse! a strong baseline for cross-lingual AMR parsing. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing* *into Enhanced Universal Dependencies (IWPT 2021)*, pages 58–64, Online. Association for Computational Linguistics.

Rik van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *CoRR*, abs/1705.09980.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2021. XLPT-AMR: Cross-lingual pre-training via multi-task learning for zero-shot AMR parsing and text generation. 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 896–907, Online. Association for Computational Linguistics.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. 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.

A Full Results and Hyper-parameters

adaptation steps	baseline	MAML
0	48.3	48.4
1	48.3	47.4
2	48.2	47.3
3	48.2	47.7
5	47.1	48.0
7	48.1	47.7
9	48.0	47.3
11	47.9	47.2
13	47.8	47.1
15	47.8	47.1

Table 6: Average SMATCH scores on target languages according to adaptation steps.

finetuning lr	baseline	MAML
0	48.4	48.3
0.0000001	48.4	48.2
0.000003	48.1	48.3
0.00001	47.3	48.2
0.00003	44.3	46.0
0.0001	31.6	36.2
0.001	21.2	19.8

Table 7: Average SMATCH scores on target languages according to fine-tuning learning rate.

	Q1	Q2	Q3	Q4	Q5	Q6
Number of languages (I)	[8, 12,14]	14	14	14	14	14
Translation source	DeepL	[DeepL, mBart]	DeepL	DeepL	DeepL	DeepL
Data size	Full	Full	[Full, 1000]	Full	Full	Full
Adpatation step (P)	0	0	0	[0, 1, 2, 3, 5, 7, 9, 11, 13, 15]	2	2
Finetuning lr rate	0	0	0	1e-5	[0, 1e-7, 3e-6, 1e-5, 3e-5, 1e-4, 1e-3]	1e-5
k **size**	0	0	0	32	32	[0, 32, 64, 128]

Table 8: Hyper-parameters settings for the research questions Q1-Q6