Cross-lingual Back-Parsing: Utterance Synthesis from Meaning Representation for Zero-Resource Semantic Parsing

Deokhyung Kang[♠], Seonjeong Hwang[♠], Yunsu Kim[♡], Gary Geunbae Lee[♠◇]
[♠]Graduate School of Artificial Intelligence, POSTECH, Republic of Korea
[◇]Department of Computer Science and Engineering, POSTECH, Republic of Korea
[♡]aiXplain, Inc. Los Gatos, CA, USA
{deokhk, seonjeongh, gblee}@postech.ac.kr, yunsu.kim@aixplain.com

Abstract

Recent efforts have aimed to utilize multilingual pretrained language models (mPLMs) to extend semantic parsing (SP) across multiple languages without requiring extensive annotations. However, achieving zero-shot cross-lingual transfer for SP remains challenging, leading to a performance gap between source and target languages. In this study, we propose **C**ross-lingual **B**ack-**P**arsing (**CBP**), a novel data augmentation methodology designed to enhance cross-lingual transfer for SP. Leveraging the representation geometry of the mPLMs, CBP synthesizes target language utterances from source meaning representations. Our methodology effectively performs cross-lingual data augmentation in challenging zero-resource settings, by utilizing only labeled data in the source language and monolingual corpora. Extensive experiments on two cross-lingual SP benchmarks (Mschema2QA and Xspider) demonstrate that CBP brings substantial gains in the target language. Further analysis of the synthesized utterances shows that our method successfully generates target language utterances with high slot value alignment rates while preserving semantic integrity.[1]

1 Introduction

Semantic Parsing (SP) is the task of converting natural language utterances into meaning representations such as SQL or Python code. With numerous English parsing datasets available, recent studies have enabled applications ranging from natural language interfaces for databases to code generation (Le et al., 2022; Li et al., 2023). Despite SP's practicality, extending it beyond English is challenging. Manually annotating examples for other languages is very costly, and relying on machine translation is often impractical due to the complex slot alignment step after translation (Nicosia et al., 2021).

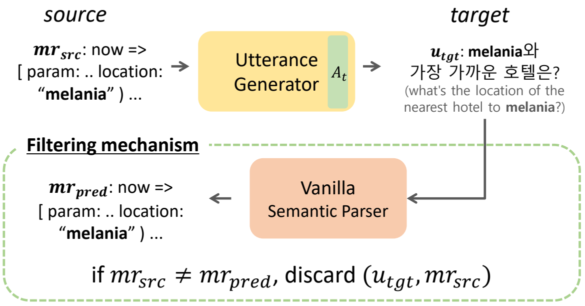

source
target
mr_{src}: now => [param: .. location: "melania") ...
Utterance Generator
A_t
u_{tgt}: melania와 가장 가까운 호텔은?
(what's the location of the nearest hotel to melania?)
Filtering mechanism
mr_{pred}: now => [param: .. location: "melania") ...
Vanilla Semantic Parser
if $mr_{src} \neq mr_{pred}$, discard (u_{tgt}, mr_{src})

Figure 1: An overview of the data augmentation process with CBP. The utterance generator equipped with the language adapter A_t synthesizes utterances in the target language t, and a filtering mechanism is applied to discard low-quality utterances. In the example, a Korean utterance is generated, with the corresponding English translation provided in parentheses.

Recent studies focus on leveraging multilingual pretrained language models (mPLMs) (Devlin et al., 2019; Xue et al., 2021) to extend SP across multiple languages without costly annotations (Sherborne and Lapata, 2022; Held et al., 2023a). After being pretrained on large-scale non-parallel multilingual corpora, mPLMs demonstrate strong zero-shot cross-lingual transferability: Once these models are fine-tuned with labeled data from the source language, they show remarkable performance in target languages without using any labeled data from the target language. Nonetheless, zero-shot cross-lingual transfer for SP is still challenging for state-of-the-art multilingual models, resulting in a notable performance gap between the source and target languages (Ruder et al., 2021).

To this end, we propose **C**ross-lingual **B**ack-**P**arsing (**CBP**), a novel data augmentation methodology for enhancing zero-shot cross-lingual transfer for SP. CBP is designed to be widely applicable by synthesizing target utterances from source meaning representations under zero-resource settings - where resources such as translators, annotated examples, and parallel corpora in target languages are

[1] Our codes and data are publicly available at https://github.com/deokhk/CBP.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 14303–14317
November 12-16, 2024 ©2024 Association for Computational Linguistics

unavailable. As shown in Figure 1, CBP comprises two components: an utterance generator synthesizing utterances in the target languages and a filtering mechanism discarding low-quality utterances.

To synthesize target utterances in the zero-resource setting, the utterance generator leverages a multilingual pretrained sequence-to-sequence (seq2seq) model such as mT5 (Xue et al., 2021) with modular language-specific adapters (Houlsby et al., 2019; Pfeiffer et al., 2020) inserted into the decoder. To enable the model to generate output text different from the input, we design a novel *source-switched denoising objective* for training the language adapters, leveraging findings (Yang et al., 2021) that the language identity component can be extracted from contextualized representations of mPLMs. Using unlabeled target language sentences, we train the adapters to denoise input sentences from encoded representations with their language identity switched to the source language. This allows the adapters to control the output language of the utterance generator during inference.

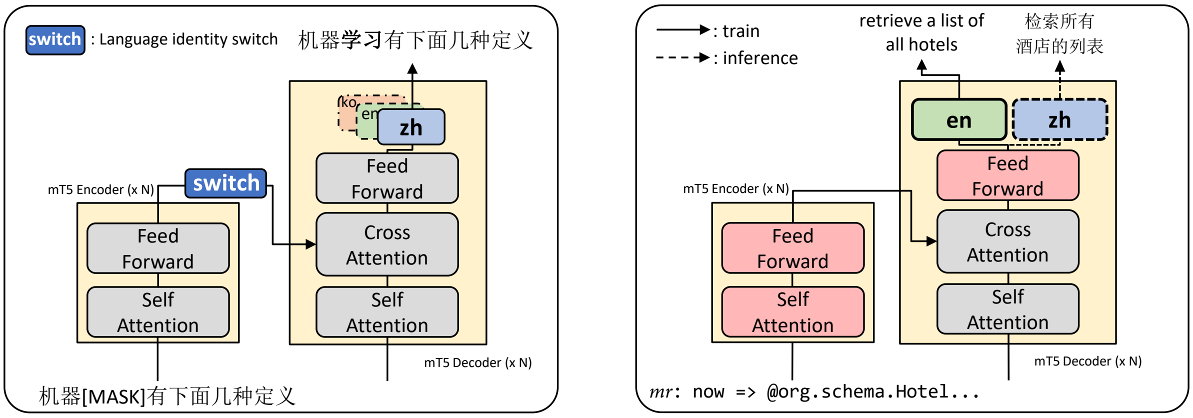
We then synthesize target utterances from the source meaning representations using the utterance generator equipped with the target language adapters. This process effectively performs data synthesis to create new target language utterances, serving as data augmentation. Finally, we filter these synthesized utterances to discard low-quality ones using a filtering mechanism inspired by round-trip consistency (Alberti et al., 2019), thereby enhancing the quality of the augmented dataset.

We assess the efficacy and robustness of CBP on two challenging cross-lingual SP benchmarks, Mschema2QA (Zhang et al., 2023) and Xspider (Shi et al., 2022), encompassing a total of 11 languages. In the Mschema2QA benchmark, CBP notably improves the average exact match by 3.2 points. Utilizing solely monolingual corpora for data augmentation, CBP surpasses all baselines that rely on translator-based data augmentation. For the Xspider benchmark, CBP exceeds the state-of-the-art, improving the exact match for Chinese from 52.7 to 54.0. Extensive analyses substantiate the effectiveness of our methodology. Further investigations into synthesized utterances indicate that CBP successfully generates utterances in the target languages high slot value alignment rates while moderately preserving semantic integrity, despite the absence of parallel corpora.

2 Related Work

Zero-shot cross-lingual semantic parsing Zero-shot cross-lingual SP aims to transfer parsing capabilities from a high-resource language (e.g., English) to low-resource languages without requiring any training data in the low-resource languages. To enhance cross-lingual transfer, several studies introduce auxiliary objectives during training to improve the alignment of semantic spaces between languages (Sherborne and Lapata, 2022; Held et al., 2023b). Our method, however, aligns with a different line of research: data augmentation. Xia and Monti (2021) utilize machine translation to convert English datasets into target languages, followed by word aligners to match corresponding elements, whereas Nicosia et al. (2021) directly generate aligned datasets using a fine-tuned model. Although not in the zero-shot setting, some works prompt large language models (LLMs) to generate synthetic data in the target language, using a few examples in the target language (Rosenbaum et al., 2022; Awasthi et al., 2023). In contrast, our research addresses data augmentation in a relatively unexplored zero-resource setting, where no target language data, translators, or parallel corpora are available. Our approach leverages multilingual pretrained language models and monolingual corpora in the target language for augmentation, ensuring effective cross-lingual transfer without such resources.

Multilingual language models Research on the representation geometry of multilingual pretrained language models (mPLMs) has revealed that the encoder representations of these models possess a shared multilingual representation space while still encoding language-specific information. A study by Libovickỳ et al. (2019) shows that subtracting the language mean from representations enhances cross-lingual transfer by inducing language-agnostic representations. Additionally, Chang et al. (2022) demonstrates that projecting representations onto language-specific subspaces can facilitate token predictions in specific languages. Leveraging these findings, Yang et al. (2021) enhances cross-lingual retrieval performance by removing language information from multilingual representations, while Deb et al. (2023) improves cross-lingual question answering by projecting source representations onto target language subspaces during fine-tuning. The study most closely related to ours is by Wu et al. (2022), which enhances



(a) Step 1: Language adapter training with **monolingual data**

(b) Step 2: utterance generation training and inference with **labeled data**

Figure 2: Overview of the utterance generator training. First, we individually train language adapters, represented by colored boxes in the decoder, using monolingual corpora for each language through source-switched denoising training. The remaining shared parameters are frozen during this process (2a). Next, we use labeled data to fine-tune the utterance generator for the utterance generation task, keeping the trained adapters frozen while selectively training the other parameters (2b). The figure is adapted from Üstün et al. (2021).

cross-lingual transfer in natural language generation tasks by selectively removing language identity during the training process of a multilingual seq2seq model. In contrast, our approach leverages the model's language identity to modify the generation language of an already fine-tuned multilingual seq2seq model, enabling cross-lingual data augmentation in a zero-resource setting.

3 Methodology

Overview In this study, we synthesize target language utterances u_{tgt} from source language meaning representation mr_{src} to enhance the performance of SP models that convert u_{tgt} into mr_{tgt}.[2] CBP consists of two components: (1) an utterance generator (Section 3.1) that synthesizes utterances in the target languages from source mr; (2) a filtering mechanism (Section 3.2) that discards low-quality synthesized utterances. To train the models in each step, we utilize SP datasets in the source language and monolingual corpora in both the source and target languages.

The utterance generator, which utilizes a seq2seq Transformer (Vaswani et al., 2017) as its backbone, is trained to generate u_{src} from the input mr_{src}, and subsequently generates u_{tgt} from mr_{src} during inference. To achieve this, the model must be capable of generating utterances in different languages from the same meaning representations. Therefore, we introduce *a language identity switch operation* and *a language-specific adapter* to control the language of the generated utterances.

The language identity switch operation alters the encoder output representation of the generator to reflect the source language identity, ensuring that the generator's decoder always receives the encoder representation with the source language identity regardless of the input language. We then train the utterance generator to produce output sequences in the target language using the modified encoder representation, while integrating language-specific adapters (Houlsby et al., 2019) into the Transformer decoder. This training enables the adapter to prompt the generator to produce utterances in different languages while maintaining the same meaning from a given representation.

Then, we remove low-quality data from the synthesized utterances using the filtering mechanism. By re-parsing the generated utterances, we measure round-trip consistency (Alberti et al., 2019) to determine whether it accurately maps back to the input meaning representation used during generation. This data filtration process improves the quality of the synthesized data.

3.1 Utterance generator

Architecture We construct the utterance generator using a multilingual pretrained seq2seq model,

[2] As shown in Figure 1, the meaning representation is largely language-independent, similar to Python code or SQL grammar, except for its slot values. Therefore, synthetic utterances in the target language that contain slot values from the source language are still useful for training SP models in target languages.

such as mT5 (Xue et al., 2021), as its backbone. To control the output languages, we integrate a language-specific adapter into each decoder block of the generator, positioning it immediately after the feed-forward layers. These adapters are lightweight bottleneck feed-forward layers that enable the generator to adapt to specific languages by learning modular representations (Pfeiffer et al., 2020; Parović et al., 2022).

Training language adapters As illustrated in Figure 2a, we initially train the language adapters using monolingual corpora for each language, respectively. Each language adapter is updated through a denoising task, where the utterance generator reconstructs randomly masked sentences into their original forms. During this training process, the model learns solely from data where the input and output sequences share the same language. However, during the data synthesis step, the model is required to generate an output sequence in the target language (u_{tgt}) when provided with an input sequence in the source language (mr_{src}). When we train the adapter with a conventional denoising objective (Lewis et al., 2020; Üstün et al., 2021), this mismatch leads to failure in synthesizing utterances in target languages (Figure 4). To mitigate this language mismatch in the zero-resource setting without parallel corpora, we propose a novel **source-switched denoising objective** to train the adapters, leveraging the representation geometry of mPLMs.

Previous studies (Libovický et al., 2020; Yang et al., 2021) have shown that the representation of mPLMs can be decomposed into language-specific and language-neutral components, which respectively capture language identity and semantic information. Inspired by this property, we **switch** the language identity of input sequences to the source language during the denoising task to prevent the model from determining the output language based on the input language. Following Libovický et al. (2020), we estimate the language-specific component for language l as the language mean vector μ_l. We compute μ_l as the mean of 1M contextualized token representations obtained from the encoder of the utterance generator, using a set of sentences from the monolingual corpora C_l.

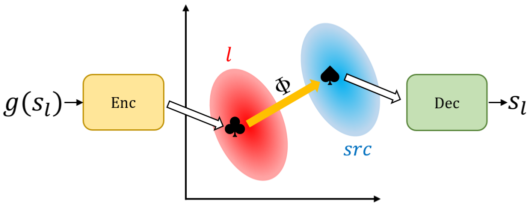


Figure 3: During source-switched denoising training, the language identity switch operation Φ switches the language identity of the encoded representation of the masked sentence ♣ from language l to the source language, resulting in ♠.

During the training of the language adapter A_l, a masked sentence $g(s_l)$ in language l is fed into the encoder Enc of the utterance generator and encoded into a representation. We then modify the language-specific component of the encoded representation to the source language using the language identity switch operation Φ. Formally, the operation is defined as:

$$\Phi(Enc(g(s_l))) = Enc(g(s_l)) - \mu_l + \mu_{src}$$

where μ_{src} is a language mean vector of the source language. This operation maintains the semantic equivalence of the representation while changing its identity to the source language (Figure 3).

The language-specific adapter learns to map input sentences from the source language to sentences in each target language while preserving the meaning, using the source-switched denoising objective. Initially, sentences are distorted using a noise function g, which replaces consecutive spans of the input sentence with a mask token. The decoder then reconstructs the original sentence based on the encoder representation with the language identity switched to the source language. For each language l, language adapter A_l is separately trained to minimize L_{A_l}:

$$L_{A_l} = \sum_{s_l \in C_l} -logP(s_l|\Phi(Enc(g(s_l))); A_l)$$

where s_l is a sentence belonging to monolingual corpora C_l of language l. All utterance generator parameters are frozen during the training except those of the adapter.

While we focus on training adapters in this work, these source-switched denoising training strategies can potentially be applied to other modular methods such as LoRA (Hu et al., 2022). We chose to focus on adapters for two main reasons: (1) they generally show better performance compared to other modular methods given the same size of trainable parameters (He et al., 2022), and (2) the literature background on their usage for cross-lingual transfer (Pfeiffer et al., 2020, 2023).

Fine-tuning Utterance generator After training language adapters for each language, we fine-tune the utterance generator to synthesize u_{src} from mr_{src} using labeled data in the source language, as shown in Figure 2b. This process involves integrating the **source** language adapter into the decoder and selectively freezing other layers of the utterance generator and the adapter to prevent catastrophic forgetting.[3]

Synthesizing target utterances After fine-tuning the utterance generator, we synthesize u_{tgt} from mr_{src}. For each mr_{src} in the labeled data, we generate u_{tgt} across various target languages by incorporating the corresponding language-specific adapter A_{tgt} into the decoder.

3.2 Filtering mechanism

To filter out low-quality synthesized utterances, we propose a filtering mechanism inspired by round-trip consistency (Alberti et al., 2019). We fine-tune the same backbone model for the utterance generator for the SP task using only labeled data from the source language. For each target language utterance u_{tgt} initially generated from mr_{src}, the trained SP model predicts its corresponding meaning representation mr_{pred}. We use the set (u_{tgt}, mr_{src}) where mr_{src} exactly matches mr_{pred} to ensure that the synthesized u_{tgt} preserves the meaning of the mr_{src}.

4 Experimental Settings

4.1 Datasets

To evaluate whether our methodology generalizes across different languages and meaning representations, we assess our methods on two cross-lingual SP datasets: Mschema2QA (Zhang et al., 2023) and Xspider (Shi et al., 2022). Examples of each dataset are presented in Table 1.

Mschema2QA is a question-answering dataset over schema.org web data that pairs user utterances with meaning representations in the ThingTalk Query Language. The dataset contains 8,932 training and 971 test examples, each available in 11 languages. Using English as the source language, we evaluate our model on the test split across 10 target languages[4].

Xspider is a cross-domain text-to-SQL dataset that pairs user utterances with SQL queries. We train our model on the English Spider dataset(Yu et al., 2018) consisting of 7,000 training examples and evaluate on the Chinese (Min et al., 2019) and Vietnamese (Nguyen et al., 2020) dev split. We did not assess Farsi and Hindi as they are not publicly available.

u	quali sono i luoghi da **piazza barberini, 9**
mr	now => (@org.schema.Hotel.Hotel) filter param:geo:Location == location: "**piazza barberini, 9**" => notify
u	那里有多少工？
mr	SELECT count(*) FROM employee

Table 1: Italian and Chinese examples of utterances (u) and corresponding meaning representations (mr) for Mschema2QA (Zhang et al. (2023), in red) and Xspider (Shi et al. (2022), in green), respectively. Mschema2QA tends to have phrase-level slot values (in bold).

Monolingual corpora We create unlabeled monolingual corpora C_l for each language l by extracting 1 million sentences from the November 20, 2023, Wikipedia dump in the respective language. We extract the raw article texts from the dump using WikiExtractor (Attardi, 2015) and split them into sentences using BlingFire (Microsoft, 2020).

4.2 Implementation details

We use the multilingual pretrained seq2seq model mT5-large (Xue et al., 2021) as the backbone for our SP model and utterance generator. The synthesized datasets for Mschema2QA and Xspider contain 49.4k and 8.2k examples, respectively. We train the model in a single stage using these synthesized datasets along with the labeled data in the source language (D_{src}), which is English. Employing AdamW (Loshchilov and Hutter, 2017) optimizer, we train the SP model for 50 epochs on both datasets, with a batch size of 32 and a learning rate of 3e-5. Appendix A.1 has further details.

4.3 Baselines

As the datasets have been proposed recently, few prior results are available in the literature. Therefore, we developed several strong baselines that do

[3] Inspired by Pfeiffer et al. (2023), we explore various freezing configurations to optimize utterance synthesis of the target language. Table 6 in the Appendix illustrates that the best results are obtained by additionally freezing the embedding layer, decoder attention, and cross-attention.

[4] Arabic (ar), German (de), Spanish (es), Persian (fa), Finnish (fi), Italian (it), Japanese (ja), Polish (pl), Turkish (tr), and Chinese (zh)

not use labeled datasets in target languages. All baselines, except those using LLM, utilize mT5-large as the backbone model.

Translation-Based Baselines For **Translate-Test**, we use Google Translate (Wu et al., 2016) to convert the target language test set into English and then input it into the model trained only with D_{src}. In **Translate-Train**, D_{src} is translated into all target languages using machine translation (MT), and a model is trained on this data. For **TAP-Train**, we translate utterances from D_{src} into all target languages with MT. Then, we use representative neural word aligners - awesome-align (Dou and Neubig, 2021) - to align utterances with values from meaning representations, constructing a dataset to train a multilingual parser. In **TAP-Train + source**, we supplement the dataset from TAP-Train with D_{src} to train the model.

In-Context Learning with Multilingual LLMs We use **gpt-3.5-turbo**[5] for in-context learning. The prompt is constructed by appending English examples and an utterance from the evaluation dataset, with eight examples for Mschema2QA and one for Xspider to meet input limits. For Xspider, we additionally compare against the state-of-the-art method that uses LLM, **DE-R^2+Translation-P** (Shi et al., 2022).

Zero-Resource Baselines For **Zero-shot**, we train a model using the English-labeled dataset D_{src} only. In **word translation**, inspired by Zhou et al. (2021), we create an augmented dataset by replacing words in English utterances from D_{src} with their counterparts in the target language, using bilingual dictionaries from MUSE (Conneau et al., 2017). To preserve alignment between the meaning representation and the utterance, we only replace words that are not part of the values. Models are trained using both D_{src} and the word-replaced dataset across target languages. For **reconstruction**, inspired by Maurya et al. (2021), we train an SP model with an auxiliary task of reconstructing input from noisy data using unlabeled corpora across target languages. This reconstruction objective aims to enrich the cross-lingual latent representation space across languages.

Additionally, We report supervised performance as an upper bound, trained on data from all languages. We train baselines utilizing mT5 with the same hyperparameters and setup as the proposed method. Additional details for baseline models can be found in Appendix A.2.

4.4 Evaluation metrics

We measure Exact Match (EM) accuracy for the Mschema2QA and XSpider datasets. Additionally, we report Test-suite (TS) accuracy for the XSpider dataset following Zhong et al. (2020). Each score is averaged over three runs with different random seeds.

5 Results and Analysis

In Tables 2 and 3, we compare the performance of CBP against competitive baselines on the Mschema2QA and Xspider benchmarks. CBP improves the average EM score on Mschema2QA by 3.2%, with significant improvements of 8.8% in Turkish and 5.0% in German, compared to the zero-shot method without data augmentation. Similarly, on Xspider, our method enhances Chinese performance by 4.7% in EM and 3.8% in TS. The filtering mechanism proves essential for our method, as evidenced by the significant drop in performance in its ablation (w/o filtering). Remarkably, despite operating under the zero-resource setting, our method outperforms all baseline models on the Mschema2QA dataset and even surpasses DE-R^2+Translation-P, the state-of-the-art in the literature on the Xspider dataset. These results highlight the effectiveness and practicality of CBP in cross-lingual SP.

Additionally, we find that gpt-3.5-turbo exhibits different performance trends on the two datasets. On Mschema2QA, gpt-3.5-turbo performs poorly, indicating that in-context learning with English examples alone is insufficient to learn the dataset's domain-specific grammar. This highlights the practicality of zero-shot cross-lingual transfer through fine-tuning. Conversely, on Xspider, where the model has pre-trained knowledge about text-to-SQL (Liu et al., 2023), gpt-3.5-turbo shows strong performance, surpassing ours in TS. However, our backbone model, mT5-large (1.2B parameters), is notably more parameter-efficient and cost-effective than gpt-3.5-turbo.

Slot value alignment One key challenge in cross-lingual data augmentation for SP is aligning slot values between the utterance and the meaning representation. Compared to translation-based baselines, we measure the slot value alignment rate of augmented data synthesized by CBP. The alignment

[5] https://platform.openai.com/docs/models/gpt-3-5-turbo

Model	ar	de	es	fa	fi	it	ja	pl	tr	zh	**Avg**
Supervised	52.8	68.1	68.9	46.7	65.9	63.5	61.9	59.0	63.7	53.7	60.4
Translate-Test	20.9	43.3	34.2	22.0	29.9	36.6	20.3	38.1	30.9	21.0	29.7
Translate-Train	13.8	40.8	37.0	17.8	32.0	33.4	2.9	37.5	31.9	7.4	25.5
TAP-Train	26.4	51.1	47.0	28.7	53.8	43.4	2.6	46.6	50.4	8.8	35.8
TAP-Train + source	28.8	60.5	53.2	29.5	54.4	48.6	5.2	47.9	55.6	8.2	39.2
gpt-3.5-turbo	13.0	15.7	15.3	11.2	16.4	15.0	7.5	15.6	14.1	12.5	13.6
Zero-shot	**32.3**	58.8	56.3	34.9	49.9	58.1	11.3	49.9	48.2	26.0	42.6
+ word translation	30.3	63.4	60.3	31.1	50.3	60.3	**13.7**	52.9	52.9	21.1	43.7
+ reconstruction	29.9	56.3	54.9	26.8	44.5	57.4	6.7	49.6	44.6	**26.2**	39.7
CBP	32.0	**63.8**	**60.4**	**35.2**	**56.8**	**62.4**	11.2	**54.8**	**57.0**	24.3	**45.8**
w/o filtering	9.4	58.4	51.3	17.1	38.0	54.6	5.5	51.3	43.9	11.0	34.0

Table 2: Exact match accuracy on Mschema2QA across (i) supervised models, (ii) translation-based models, (iii) LLM-based models, and (iv) zero-resource models. The best results among the zero-resource models are highlighted in bold.

Model	zh-full	zh		vi	
	EM	EM	TS	EM	TS
Supervised	61.3	65.9	72.2	57.4	58.5
Translate-Test	56.5	62.8	69.2	35.2	38.3
Translate-Train	57.6	63.5	70.6	47.9	50.0
TAP-Train	59.6	65.3	71.0	54.2	50.1
TAP-Train + source	59.3	64.9	69.8	53.2	50.7
gpt-3.5-turbo	38.0	35.3	67.9	37.5	55.0
DE-R^2+Translation-P[†]	47.4	52.7	55.7	43.7	43.6
Zero-shot	43.7	49.3	55.7	47.4	46.9
+ word translation	41.0	47.1	54.3	47.0	45.3
+ reconstruction	42.8	47.9	54.3	**48.6**	**47.5**
CBP	**47.8**	**54.0**	**59.5**	47.3	**47.5**
w/o filtering	41.9	47.5	54.1	41.0	42.5

Table 3: Performance on Xspider. As only a subset of data in Cspider can be evaluated with TS, we reported zh and zh-full individually, following Shi et al. (2022). † is taken from Shi et al. (2022). The best results among the zero-resource models are highlighted in bold.

rate is the percentage of training examples where the utterance contains exactly every slot value from the corresponding meaning representation. Table 4 presents each dataset's average slot value alignment rates across languages.[6] Notably, CBP consistently exhibits a **high alignment rate** across both the Xspider and Mschema2QA datasets. On the Mschema2QA dataset, our method achieves a slot alignment rate of 97.91%, significantly higher than the 75.77% achieved by the translate-and-project (TAP-Train) approach while performing competitively in Xspider.

Dataset	**Translate-Train**	**TAP-Train**	**CBP**
Mschema2QA	55.04	75.77	**97.91**
Xspider	78.89	**96.10**	94.67

Table 4: Slot value alignment rates of augmented datasets across various methods

This discrepancy can be attributed to differences in average slot value length and complexity. In Xspider, the average slot value length is 7.75 characters and 1.31 words, whereas, in Mschema2QA, it is 15.38 characters and 2.38 words, making slot alignment with word aligner (TAP-Train) more challenging. Our method, however, maintains a high slot value alignment rate in Mschema2QA, demonstrating its effectiveness in cross-lingual SP tasks with longer slot values.

Target language synthesis rate To evaluate the impact of the source-switched denoising training on synthesizing target language utterances, we assess the language of synthesized utterances in Mschema2QA using the Google Cloud Translation API's Language Detection. Figure 4 shows the target language synthesis rate. Training the language adapter with a conventional denoising objective (w/o switch) fails to synthesize target language utterances effectively. In contrast, our method, which employs a source-switched denoising objective, achieves a high synthesis rate in the target language, demonstrating its effectiveness.

Notably, our approach excels in synthesizing languages that do not share a script with the source language (English; Latin script), achieving high synthesis rates. Our method also performs robustly for languages sharing scripts, though at slightly lower rates. We speculate that shared scripts may result in similar language identities during language adapter training, potentially reducing differentia-

[6] Refer to Appendix A.3 for the process of extracting slot values and results across all target languages.

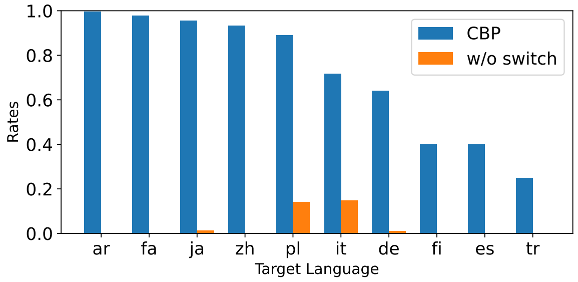


Figure 4: Target language synthesis rates (from 0 to 1) on different languages in Mschema2QA. For more granular results, refer to Appendix Table 7.

tion between languages. Nevertheless, considering the widespread use of non-Latin scripts globally, CBP's consistent target synthesis rate with these scripts highlights its broad applicability and effectiveness.

Quality of synthesized utterances We evaluate the translation quality between the synthesized utterance from CBP and the English utterance paired with the meaning representation for the synthesized utterance. We assess the quality only for the synthesized utterances that were identified as being in the target language by the language detection API. We employ GEMBA-stars (Kocmi and Federmann, 2023), a state-of-the-art GPT-based metric that assesses translation quality on a one-to-five-star scale through zero-shot prompting. Figure 5 shows the star distribution for synthesized utterances across all languages on Mschema2QA. We find that the majority of utterances fall within the two to four-star range, indicating similar meaning to some degree. This suggests that our method not only adjusts the synthesized utterances' language but also preserves their meaning to some extent. Synthesized utterances across different languages are presented in Figure 10 in the Appendix.

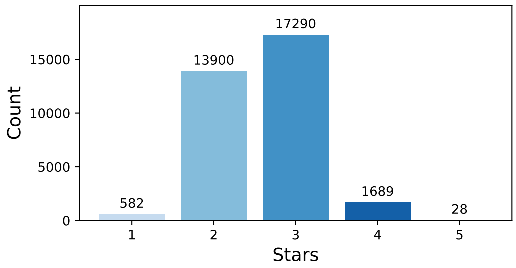


Figure 5: Quality of synthesized utterances measured by GEMBA-stars. We use gpt-3.5-turbo as the backbone.

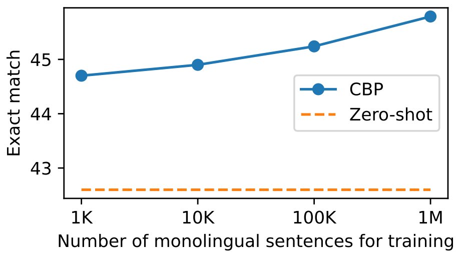


Figure 6: Average exact match on Mschema2QA across different languages with varying monolingual corpora sizes (1K to 1M)

Monolingual data size To assess the impact of monolingual corpora size in source-switched denoising training, we trained adapters for the languages used in Mschema2QA with progressively smaller data sizes. Figure 6 illustrates the average performance of Mschema2QA when trained with data sizes ranging from 1K to 1M across different languages. The results reveal better performance with more data, but notably, CBP outperformed the zero-shot method even with just 1K corpora per language. This demonstrates that our approach can be effective even for languages where acquiring large monolingual corpora is challenging.

Impact of zero-shot SP performance Figure 7 illustrates the relationship between zero-shot EM performance and improvement through our data augmentation across various languages. The results show that languages with higher zero-shot performance tend to exhibit greater improvements from data augmentation. A notable comparison can

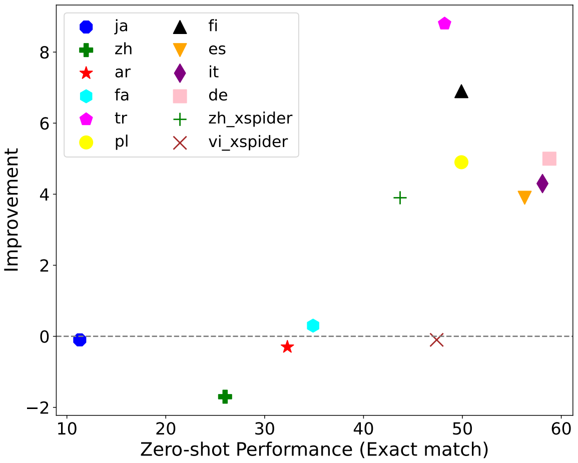


Figure 7: Relationship between zero-shot exact match performance and improvement through data augmentation (in exact match) across various languages. For those without the suffix "xspider," it pertains to mschema2qa.

be made between zh and zh_xspider. Despite both being in Chinese, zh_xspider, which has higher zero-shot task performance than zh, demonstrates significant gains from data augmentation, whereas zh does not. This indicates that the zero-shot performance of the task, rather than the language itself, is the primary factor influencing the magnitude of improvement through data augmentation. We hypothesize this is due to our method's use of zero-shot cross-lingual transferability of mPLMs for data synthesis and filtering, making initial zero-shot performance crucial. Future research could focus on enhancing data augmentation techniques to work effectively for languages with low zero-shot task performance.

6 Conclusion

We present Cross-lingual Back-Parsing (CBP), a novel data augmentation methodology aimed at enhancing zero-shot cross-lingual transfer for semantic parsing. Leveraging the representation geometry of multilingual pretrained language models, our method enables data augmentation in zero-resource settings. Our experiments on two cross-lingual semantic parsing benchmarks demonstrate that CBP significantly improves performance, underscoring its effectiveness and practical applicability. While we focus on semantic parsing, we believe that CBP has the potential to be applied to other cross-lingual generation tasks in zero-resource settings. Future work will investigate the application of our method to tasks such as cross-lingual text style transfer (Krishna et al., 2022).

7 Limitations

Our proposed methodology, CBP, synthesizes target language utterances from source meaning representations by leveraging the representation geometry of mPLMs. Although we have demonstrated that CBP can effectively synthesize target utterances while preserving semantics, our experiments were conducted using only one mPLM (mT5-large). Validating our methodology with mPLMs of different parameter sizes and pretraining objectives would further demonstrate its generalizability. Additionally, while we demonstrated that our approach is beneficial even when the available monolingual corpora are small in size (Figure 6; applicable to actual low-resource language settings), we couldn't experiment on actual low-resource languages due to the limited natural language coverage of current semantic parsing datasets (Zhang et al., 2023). Evaluating our methodology on actual low-resource languages could further verify its effectiveness. Finally, our methodology is less effective in synthesizing data when the zero-shot task performance is low. This indicates that our approach may not be effective for mPLMs with lower inherent performance, such as small-sized models. Future work could focus on improving our methodology to enhance performance even in these challenging scenarios.

Acknowledgements

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2024-2020-0-01789) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation), and by Hyundai Mobis. We thank the anonymous reviewers for their helpful comments and suggestions.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173.

Giusepppe Attardi. 2015. Wikiextractor. `https://github.com/attardi/wikiextractor`.

Abhijeet Awasthi, Nitish Gupta, Bidisha Samanta, Shachi Dave, Sunita Sarawagi, and Partha Talukdar. 2023. Bootstrapping multilingual semantic parsers using large language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2455–2467.

Tyler Chang, Zhuowen Tu, and Benjamin Bergen. 2022. The geometry of multilingual language model representations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 119–136.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Ujan Deb, Ridayesh Parab, and Preethi Jyothi. 2023. Zero-shot cross-lingual transfer with learned projections using unlabeled target-language data. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short*

Papers), pages 449–457, Toronto, Canada. Association for Computational Linguistics.

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

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

William Held, Christopher Hidey, Fei Liu, Eric Zhu, Rahul Goel, Diyi Yang, and Rushin Shah. 2023a. DAMP: Doubly aligned multilingual parser for task-oriented dialogue. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3586–3604, Toronto, Canada. Association for Computational Linguistics.

William Held, Christopher Hidey, Fei Liu, Eric Zhu, Rahul Goel, Diyi Yang, and Rushin Shah. 2023b. Damp: Doubly aligned multilingual parser for task-oriented dialogue. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3586–3604.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. *arXiv preprint arXiv:2302.14520*.

Kalpesh Krishna, Deepak Nathani, Xavier Garcia, Bidisha Samanta, and Partha Talukdar. 2022. Few-shot controllable style transfer for low-resource multilingual settings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7439–7468.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. 2022. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:21314–21328.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Haoyang Li, Jing Zhang, Cuiping Li, and Hong Chen. 2023. Resdsql: Decoupling schema linking and skeleton parsing for text-to-sql. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13067–13075.

Jindřich Libovickỳ, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *arXiv preprint arXiv:1911.03310*.

Jindřich Libovickỳ, Rudolf Rosa, and Alexander Fraser. 2020. On the language neutrality of pre-trained multilingual representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1663–1674.

Aiwei Liu, Xuming Hu, Lijie Wen, and Philip S Yu. 2023. A comprehensive evaluation of chatgpt's zero-shot text-to-sql capability. *arXiv preprint arXiv:2303.13547*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Kaushal Kumar Maurya, Maunendra Sankar Desarkar, Yoshinobu Kano, and Kumari Deepshikha. 2021. ZmBART: An unsupervised cross-lingual transfer framework for language generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2804–2818, Online. Association for Computational Linguistics.

Microsoft. 2020. Blingfire. `https://github.com/microsoft/BlingFire`.

Qingkai Min, Yuefeng Shi, and Yue Zhang. 2019. A pilot study for chinese sql semantic 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 3652–3658.

Anh Tuan Nguyen, Mai Hoang Dao, and Dat Quoc Nguyen. 2020. A Pilot Study of Text-to-SQL Semantic Parsing for Vietnamese. In *Findings of the*

Association for Computational Linguistics: EMNLP 2020, pages 4079–4085.

Massimo Nicosia, Zhongdi Qu, and Yasemin Altun. 2021. Translate & fill: Improving zero-shot multilingual semantic parsing with synthetic data. *arXiv preprint arXiv:2109.04319*.

Marinela Parović, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2022. Bad-x: Bilingual adapters improve zero-shot cross-lingual transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1791–1799.

Jonas Pfeiffer, Francesco Piccinno, Massimo Nicosia, Xinyi Wang, Machel Reid, and Sebastian Ruder. 2023. mmT5: Modular multilingual pre-training solves source language hallucinations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1978–2008, Singapore. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673.

Andy Rosenbaum, Saleh Soltan, Wael Hamza, Marco Damonte, Isabel Groves, and Amir Saffari. 2022. Clasp: Few-shot cross-lingual data augmentation for semantic parsing. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 444–462.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, et al. 2021. Xtreme-r: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245.

Tom Sherborne and Mirella Lapata. 2022. Zero-shot cross-lingual semantic parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4134–4153.

Peng Shi, Rui Zhang, He Bai, and Jimmy Lin. 2022. Xricl: Cross-lingual retrieval-augmented in-context learning for cross-lingual text-to-sql semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5248–5259.

Ahmet Üstün, Alexandre Bérard, Laurent Besacier, and Matthias Gallé. 2021. Multilingual unsupervised neural machine translation with denoising adapters. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6650–6662.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xianze Wu, Zaixiang Zheng, Hao Zhou, and Yong Yu. 2022. Laft: Cross-lingual transfer for text generation by language-agnostic finetuning. In *Proceedings of the 15th International Conference on Natural Language Generation*, pages 260–266.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Menglin Xia and Emilio Monti. 2021. Multilingual neural semantic parsing for low-resourced languages. In *Proceedings of* SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 185–194.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Ziyi Yang, Yinfei Yang, Daniel Cer, and Eric Darve. 2021. A simple and effective method to eliminate the self language bias in multilingual representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5825–5832.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Yusen Zhang, Jun Wang, Zhiguo Wang, and Rui Zhang. 2023. Xsemplr: Cross-lingual semantic parsing in multiple natural languages and meaning representations. In *ACL*.

Ruiqi Zhong, Tao Yu, and Dan Klein. 2020. Semantic evaluation for text-to-sql with distilled test suites. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 396–411.

Yucheng Zhou, Xiubo Geng, Tao Shen, Wenqiang Zhang, and Daxin Jiang. 2021. Improving zero-shot cross-lingual transfer for multilingual question answering over knowledge graph. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5822–5834, Online. Association for Computational Linguistics.

A Appendix

A.1 Implementation details

Language adapter We initialize the language adapter as a bottleneck feed-forward layer, following the configuration from (Pfeiffer et al., 2020). We train the adapter individually on source-switched denoising objectives using 1M sentences extracted from Wikipedia in each target language. We train the model for 100K steps with a batch size of 32 and a learning rate of 1e-4. We use AdamW (Loshchilov and Hutter, 2017) as an optimizer and minimize cross-entropy loss between the predicted sentence and the ground truth sentence. We use span masking as the noise function g, following the pretraining approach of mBART (Liu et al., 2020). A span of tokens is substituted with the mask token, with its length randomly sampled by a Poisson distribution with lambda=3.5. The training process took 26 hours on four A100-80GB GPUs.

Question generator We initialize the question generator with mT5-large (Xue et al., 2021). We train the model for 50 epochs on both Mschema2QA and Xspider, with a batch size of 4 and a learning rate of 3e-5. We use AdamW (Loshchilov and Hutter, 2017) as an optimizer, and we minimize cross-entropy loss between the predicted question and the ground truth question. We used the final checkpoint to minimize errors during the utterance generation process. For Xspider, we append linearized databases to the input, following the style used by Li et al. (2023). The training process took 8 hours for Xspider and 6 hours for Mschema2QA with four A100-80GB GPUs.

Semantic parser We initialize the semantic parser with mT5-large (Xue et al., 2021). We train the model for 50 epochs on both Mschema2QA and Xspider, with a batch size of 32 and a learning rate of 3e-5. We use AdamW (Loshchilov and Hutter, 2017) as an optimizer, and minimize cross-entropy loss between the predicted meaning representation and the ground truth representation. We select checkpoints based on validation performance on the English dev set. For Xspider, we append linearized databases to the input, following the style used by Li et al. (2023). The training process took 5 hours for Xspider and 11 hours for MschemaQA with four RTX6000ADA GPUs.

A.2 Additional details for baseline models

In this section, we provide additional details for the following baselines: TAP-Train, gpt-3.5-turbo, and reconstruction.

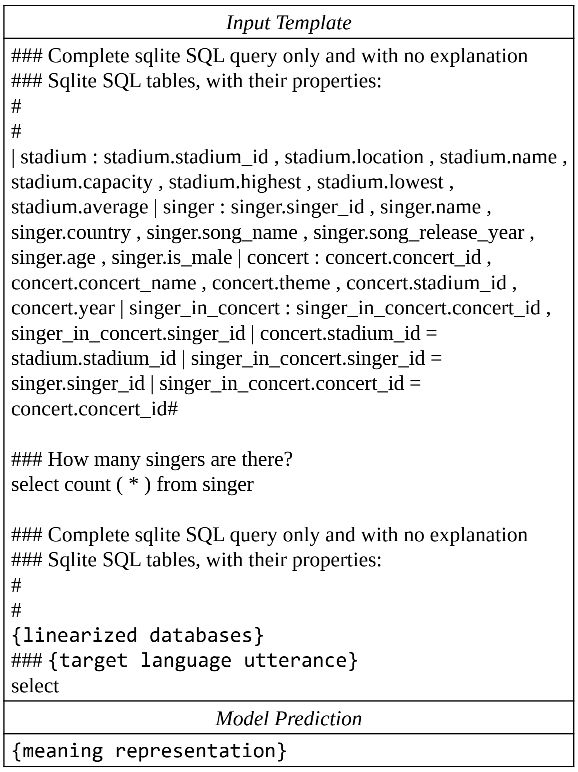
TAP-Train Both Mschema2QA and Xspider have slot values in their meaning representations enclosed in double quotes. We used regex to extract these slot values. Using the unsupervised version of awesome-align (Dou and Neubig, 2021), we replaced the slot values in the meaning representation with the corresponding values in the utterance. We found that replacing values in the utterance with the corresponding slot values from the meaning representation performed worse, so we opted to replace the slot values in the meaning representation instead.

GPT-3.5-turbo We construct the prompt for gpt-3.5-turbo by appending English examples and an utterance in the target language from the evaluation dataset. To meet input limits, we include eight examples for Mschema2QA and one for Xspider, selected randomly. Figures 8 and 9 show the prompt format for Mschema2QA and Xspider, respectively. We used regular expressions to post-process the model's predictions, extracting only the required meaning representations.

Input Template
Translate the following question into thingtalk QL: is there michelin 1 star red lion hotel monterey lodgings which have location where i am now MR: now => (@org.schema.Hotel.Hotel) filter param:geo:Location == location:current_location and param:id:Entity(org.schema.Hotel:Hotel) =~ " red lion hotel monterey " and param:starRating.ratingValue:Number == 1 => notify ··· *seven English examples omitted for brevity* ··· Translate the following question into thingtalk QL: `{target language utterance}` MR:
Model Prediction
`{meaning representation}`

Figure 8: The input and output template for few-shot inference of GPT-3.5-turbo for Mschema2QA

Reconstruction In addition to the cross-entropy loss L_{SP} used for semantic parsing training, we introduce a loss L_{RE} for the auxiliary task of reconstructing input from noisy data using unlabeled corpora across target languages. We utilized the same unlabeled corpora (Wikipedia) that were employed



Input Template

```
### Complete sqlite SQL query only and with no explanation
### Sqlite SQL tables, with their properties:
#
#
| stadium : stadium.stadium_id , stadium.location , stadium.name ,
stadium.capacity , stadium.highest , stadium.lowest ,
stadium.average | singer : singer.singer_id , singer.name ,
singer.country , singer.song_name , singer.song_release_year ,
singer.age , singer.is_male | concert : concert.concert_id ,
concert.concert_name , concert.theme , concert.stadium_id ,
concert.year | singer_in_concert : singer_in_concert.concert_id ,
singer_in_concert.singer_id | concert.stadium_id =
stadium.stadium_id | singer_in_concert.singer_id =
singer.singer_id | singer_in_concert.concert_id =
concert.concert_id#

### How many singers are there?
select count ( * ) from singer

### Complete sqlite SQL query only and with no explanation
### Sqlite SQL tables, with their properties:
#
#
{linearized databases}
### {target language utterance}
select
```

Model Prediction

```
{meaning representation}
```

Figure 9: The input and output template for few-shot inference of GPT-3.5-turbo for Xspider. We used the prompt format in (Liu et al., 2023), while linearizing databases following styles used by (Li et al., 2023).

to train the language adapter. After extracting sentences from these corpora, we applied the identical noise function used in the language adapter training, which masks spans of tokens. The auxiliary task aims to reconstruct the original input from this noised input, and we utilized the cross-entropy loss between the predicted input and the original input for the task. The final loss is computed as follows:

$$L = L_{SP} + \alpha L_{RE},$$

where α is the weight for L_{RE}. We empirically optimized α to 0.01 among candidates of [0.001, 0.01, 0.1, 0.5], as it performed the best in the evaluation.

A.3 Slot value alignment results across languages

Mschema2QA and Xspider include slot values in their meaning representations, enclosed in double quotes. We used regex to extract these slot values. We measure the slot value alignment rate as the percentage of examples where the utterance contains every slot value (EM) in the corresponding meaning representation. In cases where there are no slot values, we consider the alignment to be satisfied. To assess the alignment rate's impact on the model's performance with the augmented dataset, we compute the alignment rate across examples in the training set for each dataset. Table 5 presents the slot value alignment rates across languages in Mschema2QA and Xspider.

	Mschema2QA										Xspider	
	ar	de	es	fa	fi	it	ja	pl	tr	zh	zh	vi
Translate-Train	44.02	67.28	63.95	41.49	61.79	68.06	38.98	58.32	58.80	47.69	76.04	81.74
TAP-Train	72.76	80.59	77.79	70.09	81.73	80.5	68.18	76.57	85.17	64.33	**94.56**	**97.63**
CPB	**98.69**	**98.78**	**99.03**	**97.75**	**99.21**	**99.09**	**92.16**	**98.74**	**98.9**	**96.76**	93.15	96.18

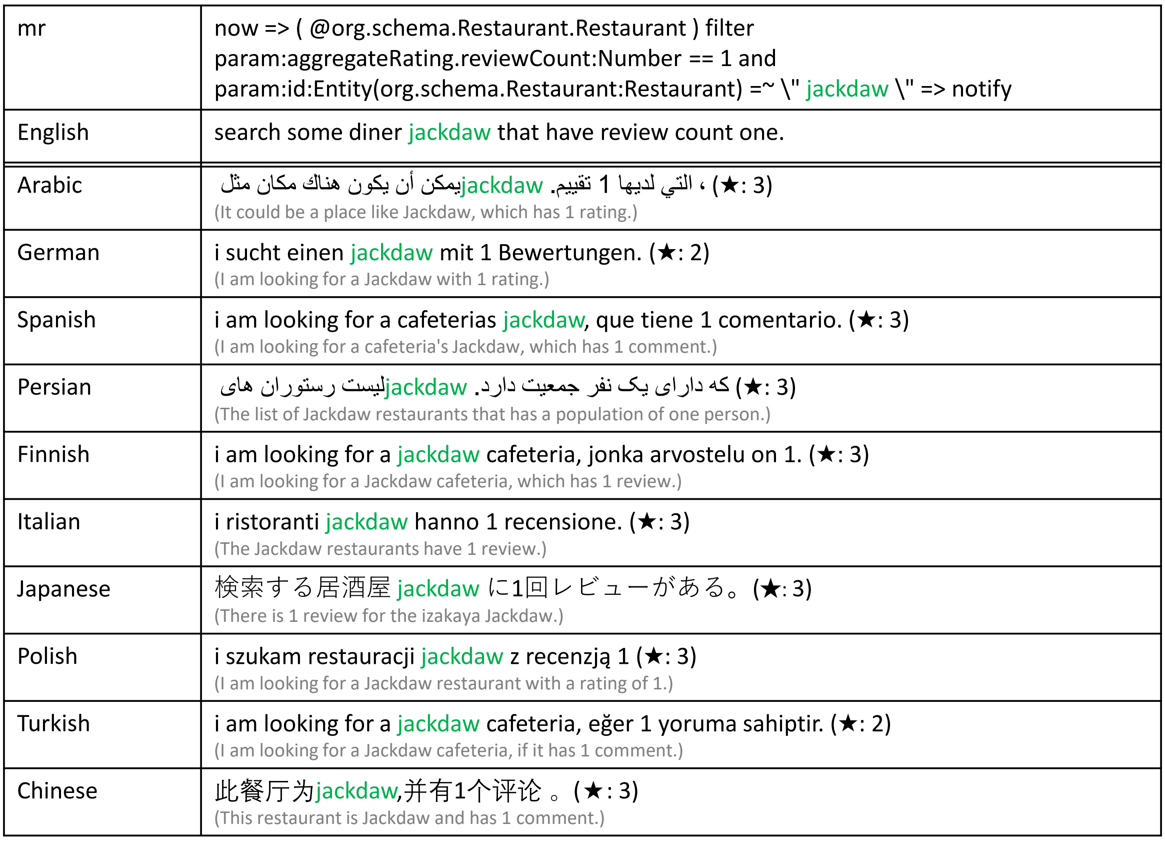
Table 5: Slot value alignment results across languages. The best results are in bold.

Emb	Enc$_{LN}$	Enc$_{Att}$ + Enc$_{FFN}$	Dec$_{LN}$	Dec$_{att}$	Dec$_{catt}$	Dec$_{FFN}$	Target synthesis rate		
							zh	vi	Avg.
							51.2	6.2	28.7
X	X	X					50.5	13.9	32.2
X			X	X	X	X	93.3	15.6	54.4
X				X	X		88.0	27.7	**57.8**
X	X		X	X		X	75.0	11.6	43.3
X	X		X			X	84.9	6.8	45.8

Table 6: Utterance synthesis rate in the target language of different freezing configurations, measured in Xspider. "X" denotes a frozen component.

Method	Synthesized Language	Mschema2QA										Xspider	
		ar	de	es	fa	fi	it	ja	pl	tr	zh	zh	vi
w/o switch	target	0.00	0.00	1.18	0.00	13.99	14.18	0.92	0.00	0.00	0.00	0.02	10.65
	source (en)	99.82	98.64	96.86	98.86	85.47	84.38	98.84	99.62	99.36	99.71	99.87	89.33
	others	0.18	1.36	1.96	1.14	0.54	1.44	0.24	0.38	0.64	0.29	0.11	0.02
CBP	target	99.58	64.10	39.93	97.75	40.15	71.73	95.52	89.01	24.82	93.34	88.02	27.65
	source (en)	0.30	35.42	57.82	1.93	59.17	26.32	3.84	10.88	74.32	6.23	11.37	72.35
	others	0.12	0.48	2.25	0.32	0.68	1.95	0.64	0.11	0.86	0.43	0.61	0.00

Table 7: Language distribution of synthesized utterances using language adapters trained with different methods



mr	now => (@org.schema.Restaurant.Restaurant) filter param:aggregateRating.reviewCount:Number == 1 and param:id:Entity(org.schema.Restaurant:Restaurant) =~ \" jackdaw \" => notify
English	search some diner jackdaw that have review count one.
Arabic	يمكن أن يكون هناك مكان مثل jackdaw، التي لديها 1 تقييم. (★: 3) (It could be a place like Jackdaw, which has 1 rating.)
German	i sucht einen jackdaw mit 1 Bewertungen. (★: 2) (I am looking for a Jackdaw with 1 rating.)
Spanish	i am looking for a cafeterias jackdaw, que tiene 1 comentario. (★: 3) (I am looking for a cafeteria's Jackdaw, which has 1 comment.)
Persian	لیست رستوران های jackdaw که دارای یک نفر جمعیت دارد. (★: 3) (The list of Jackdaw restaurants that has a population of one person.)
Finnish	i am looking for a jackdaw cafeteria, jonka arvostelu on 1. (★: 3) (I am looking for a Jackdaw cafeteria, which has 1 review.)
Italian	i ristoranti jackdaw hanno 1 recensione. (★: 3) (The Jackdaw restaurants have 1 review.)
Japanese	検索する居酒屋 jackdaw に1回レビューがある。(★: 3) (There is 1 review for the izakaya Jackdaw.)
Polish	i szukam restauracji jackdaw z recenzją 1 (★: 3) (I am looking for a Jackdaw restaurant with a rating of 1.)
Turkish	i am looking for a jackdaw cafeteria, eğer 1 yoruma sahiptir. (★: 2) (I am looking for a Jackdaw cafeteria, if it has 1 comment.)
Chinese	此餐厅为jackdaw,并有1个评论 。(★: 3) (This restaurant is Jackdaw and has 1 comment.)

Figure 10: An example of synthesized utterances generated from a meaning representation of Mschema2QA (Zhang et al., 2023). The synthesized utterances for each language are presented along with their corresponding English translations in parentheses. The numbers next to the stars indicate translation quality measured by GEMBA-stars (Kocmi and Federmann, 2023). The synthesized utterances convey a meaning similar to that of the English utterance. Additionally, the slot values remain unchanged in the synthesized utterances (in green).