Bootstrapping Multilingual Semantic Parsers using Large Language Models

Abhijeet Awasthi[1*] **Nitish Gupta**[2] **Bidisha Samanta**[2]
Shachi Dave[2] **Sunita Sarawagi**[1] **Partha Talukdar**[2]

[1]Indian Institute of Technology Bombay, [2]Google Research India
{awasthi,sunita}@cse.iitb.ac.in
{guptanitish,bidishasamanta,shachi,partha}@google.com

Abstract

Despite cross-lingual generalization demonstrated by pre-trained multilingual models, the translate-train paradigm of transferring English datasets across multiple languages remains to be a key mechanism for training task-specific multilingual models. However, for many low-resource languages, the availability of a reliable translation service entails significant amounts of costly human-annotated translation pairs. Further, translation services may continue to be brittle due to domain mismatch between task-specific input text and general-purpose text used for training translation models. For multilingual semantic parsing, we demonstrate the effectiveness and flexibility offered by large language models (LLMs) for translating English datasets into several languages via few-shot prompting. Through extensive comparisons on two public datasets, MTOP and MASSIVE, spanning 50 languages and several domains, we show that our method of translating data using LLMs outperforms a strong translate-train baseline on 41 out of 50 languages. We study the key design choices that enable more effective multilingual data translation via prompted LLMs.

1 Introduction

Enabling language technologies across several languages is an important goal for serving a diverse range of users in an inclusive manner. Recent advances in large-scale self-supervised multilingual language models hold immense promise in bridging the quality gap that currently exists between English and many other low resource languages (Conneau et al., 2020; Brown et al., 2020; Xue et al., 2021). Even though multilingual models exhibit cross-lingual generalization, getting meaningful performance across several languages still requires significant amounts of task-specific labeled data.

We consider the problem of automatically synthesizing semantic parsing datasets across several languages. Semantic parsing (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Berant et al., 2013) is the task of mapping natural language text into an executable *logical-form*. For example, given a user instruction (x) : "`Wake me up by 5 am`", mapping it to the logical-form (y): `[IN:CREATE_ALARM [SL:DATE_TIME 5 am ]]`. Manual annotation of queries with their logical forms requires human expertise which makes data collection across multiple languages challenging.

A common approach to automatic multilingual dataset creation is translating existing English datasets into target languages. Prior methods utilize an off-the-shelf machine translation model for translating the English utterance into the target language $x_{\text{eng}} \rightarrow x_{\text{tgt}}$, followed by projecting language specific components in the English logical-form y_{eng} to obtain the logical-form y_{tgt} in the target language (Moradshahi et al., 2020, 2021; Xia and Monti, 2021; Nicosia et al., 2021; Gritta et al., 2022; Wang et al., 2022). The projection step is often learned independent of the translation service, resulting in poor generalization across languages.

In this work we aim to utilize the few-shot generalization abilities exhibited by large language models (LLMs) (Brown et al., 2020; Chowdhery et al., 2022; Scao et al., 2022) for bootstrapping semantic parsing datasets across fifty languages. We propose a recipe of using LLMs to translate an English semantic parsing dataset containing `(utterance, logical-form)` pairs: $\mathcal{D}_{\text{eng}} = \{(x^i_{\text{eng}}, y^i_{\text{eng}})\}$ into a corresponding dataset in a target language: $\mathcal{D}_{\text{tgt}} = \{(x^i_{\text{tgt}}, y^i_{\text{tgt}})\}$. The generated dataset $\mathcal{D}_{\text{tgt}}$ is then used to train a semantic parser in the target language. Our method uses a small amount of manually translated semantic parsing examples to *teach* the LLM how to translate English examples in the target language via in-context learning (Min et al., 2022).

Figure 1 describes our data-translation pipeline which we refer to as LLM-T (§ 3). In contrast

[*]Work done during an internship at Google Research

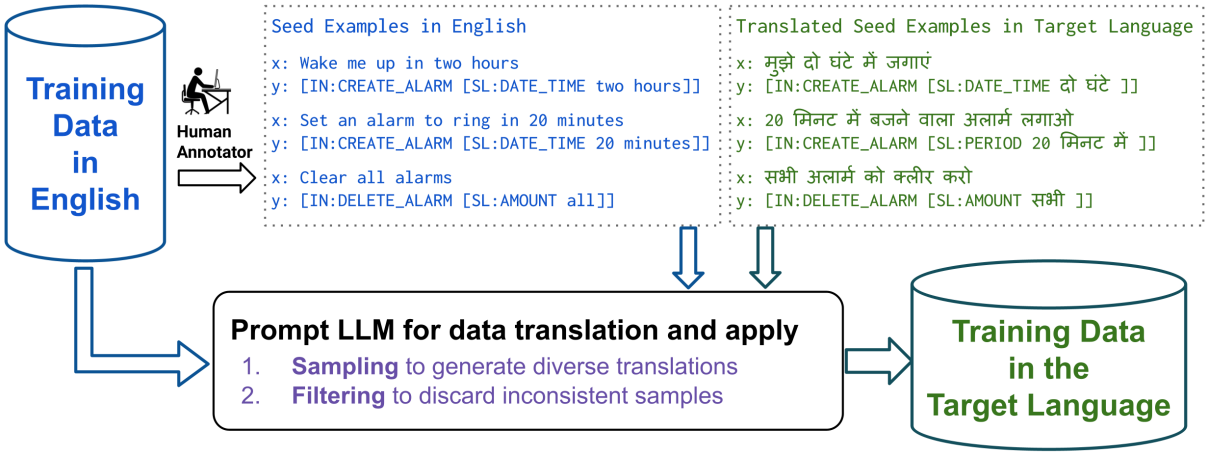

Figure 1: **Proposed semantic parsing data translation pipeline using LLMs** (§ 3): With the help of human translators, we first collect translations of a small seed set of English examples in the Target Language (e.g. Hindi; § 3.1). Given a new English example, a small subset from this initial seed set of examples with their respective translations is chosen to prompt the LLM (§ 3.2). The prompted LLM translates the given English example in the Target Language. We repeat this process for each example in the English training data to generate a training dataset in the Target Language. To ensure high-quality of the resulting dataset, we generate diverse translations via top-p (nucleus) sampling (§ 3.3) and apply consistency filtering (§ 3.4).

to prior translation based methods that involved a two-staged process requiring different modules, our method uses the LLM to *jointly translate* an English $(x_{\text{eng}}, y_{\text{eng}})$ pair directly into the target language $(x_{\text{tgt}}, y_{\text{tgt}})$. We identify two important choices that make the LLM translated data more effective for training a downstream parser: **(i) Sampling diverse translations** (§ 3.3): Decoding translations using top-p (Fan et al., 2018) and top-k (Holtzman et al., 2019) sampling leads to improved downstream performance compared to using greedy decoding. Sampling multiple diverse translations per example further improves the downstream performance; **(ii) Filtering inconsistent examples** (§ 3.4): Decoding via sampling can result in noisy joint translations of the (utterance, logical-form) pairs. To filter out the inconsistent pairs, we propose a slot-value match based filtering technique that improves the training data quality.

We perform experiments on two multilingual semantic parsing datasets: MTOP (Li et al., 2021) and MASSIVE (FitzGerald et al., 2022). On *4 out of 5* languages in MTOP and *41 out of 50* languages in MASSIVE, our method LLM-T outperforms TAF (Nicosia et al., 2021), a strong baseline that utilizes a supervised translation service (§ 5.1). Further, we see that LLM-T achieves 93% of the performance obtained by "fully-supervised" models that use 30× more manually translated examples (§ 5.2). We justify the importance of generating multiple translations using sampling, filtering out inconsistent examples, and using larger-sized LLMs in improving translated data quality (§ 5.3). Finally, we perform an error analysis of our parser and show the key sources of disagreements between the model predictions and the ground truth (§ 5.4).

2 Background

In this section, we provide an overview of semantic parsing and prior translation-based methods for creating multilingual semantic parsing datasets.

2.1 Semantic Parsing

Semantic parsing is the task of mapping text queries to their meaning representations or *logical forms* (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Berant et al., 2013). We focus on task-oriented semantic parsing (Gupta et al., 2018) where the user utterance needs to be parsed into a high-level intent specifying the overall goal, and fine-grained slots containing details about the utterance. The intents and slots come from a task-specific vocabulary. For example, given an utterance x: "How is the rainfall today?", the parser should generate the logical-form y: [IN:GET_WEATHER [SL:ATTRIBUTE rainfall] [SL:DATE today]]

Here, IN:GET_WEATHER is the high-level intent, SL:ATTRIBUTE and SL:DATE are the slots that specify details about the intent. We refer to the logical-

form with its slot values removed as its "signature". For example, the signature of y is `[IN:GET_WEATHER [SL:ATTRIBUTE][SL:DATE]]`

2.2 Translating Semantic Parsing Datasets

Given an English semantic parsing dataset containing (utterance, logical-form) pairs $\mathcal{D}_{\text{eng}} = \{(x^i_{\text{eng}}, y^i_{\text{eng}})\}$, many methods aim to translate $\mathcal{D}_{\text{eng}}$ to a dataset $\mathcal{D}_{\text{tgt}} = \{(x^i_{\text{tgt}}, y^i_{\text{tgt}})\}$ in the target language (tgt). Here x^i_{tgt} is the translation of x^i_{eng}, and y^i_{tgt} is the logical form grounded in the translated utterance x^i_{tgt}. Target logical form y^i_{tgt} has the same signature as y^i_{eng} and only differs in terms of the translated slot values. Most translation based approaches (Moradshahi et al., 2020, 2021; Xia and Monti, 2021; Nicosia et al., 2021) translate an English example $(x^i_{\text{eng}}, y^i_{\text{eng}})$ to the corresponding target language example $(x^i_{\text{tgt}}, y^i_{\text{tgt}})$ via a two step process: (i) **Translate**: Use a supervised translation service to convert the English utterance x^i_{eng} into the target language utterance x^i_{tgt}; and (ii) **Project**: Replace the English slot values in y^i_{eng} with spans copied from the translated utterance x^i_{tgt} via a learned alignment model. The translated examples are then used to train a downstream multilingual semantic parser. For example, Nicosia et al. (2021) implement the project step by training a filler module on English data to fill slot-values in a logical-form signature by copying spans from the utterance. During inference, the trained filler module is then used in a zero-shot manner to fill logical-form signatures with spans copied from the translated utterances.

3 Our Method: Prompting LLMs for Dataset Translation

Our goal is to learn a multilingual semantic parser capable of parsing user queries in many languages. Towards this goal, we propose a method for generating multilingual training datasets via few-shot prompting of an LLM to translate existing English datasets into several languages.

In contrast to prior approaches, we jointly perform example translation by prompting an LLM with a few exemplars of translating English $(x_{\text{eng}}, y_{\text{eng}})$ pairs to target language $(x_{\text{tgt}}, y_{\text{tgt}})$ pairs. Figure 1 describes our data-translation method which we refer to as LLM-T. With the help of human translators we first collect a small seed set of exemplar translations used for prompting the LLM (§ 3.1). Given an input English example, we dynamically construct the LLM prompt by identifying a relevant subset of seed exemplars (§ 3.2). The LLM translates the English example into the target language by in-context learning from the exemplars provided in the prompt. Instead of decoding the most likely translation, we generate multiple diverse translations (§ 3.3) using top-p (nucleus) sampling (Holtzman et al., 2019). While sampling improves the text diversity, it can lead to more noisy generations. We filter out the noisy generations using a simple string-match based technique before training a parser on the translated data (§ 3.4).

3.1 Selecting Seed Exemplars for Translation

Given an English semantic parsing dataset $\mathcal{D}_{\text{eng}} = \{(x^i_{\text{eng}}, y^i_{\text{eng}})\}$, we first want to identify a small seed set $\mathcal{S}_{\text{eng}} \subset \mathcal{D}_{\text{eng}}$ that will be translated into the target language ($\mathcal{S}_{\text{tgt}}$) with the help of human translators. The examples in $\mathcal{S}_{\text{eng}}$ and their corresponding translations in $\mathcal{S}_{\text{tgt}}$ will be used for prompting the LLM. Therefore, the choice of the seed examples in $\mathcal{S}_{\text{eng}}$ that are manually translated into $\mathcal{S}_{\text{tgt}}$ becomes important—we would like that the multiple domains (e.g. `Alarms, Music, News, Weather, etc.`) and the intents and slot types in each domain are covered. This ensures that for a given English example to be translated, we will be able to prompt the LLM in a manner such that at least one of the few-shot exemplars will share the intent and slots with the test English example. In practice, we select seed examples in a manner to cover all the intents and slots in a domain at least once. If the selected examples are less than 20 for a domain, we select the remaining examples randomly.

3.2 Constructing the Prompt using Translation Pairs in the Seed Sets

LLM inference is constrained by the maximum number of tokens in the input. Hence, we can only fit a limited number of examples to construct the LLM prompt. The choice of prompt examples and their ordering is known to significantly impact the quality of the generations (Kumar and Talukdar, 2021; Rubin et al., 2021; Lu et al., 2022). To improve the likelihood of correctly translating an English example $(x_{\text{eng}}, y_{\text{eng}})$, we retrieve seed examples $\{(x^s_{\text{eng}}, y^s_{\text{eng}}, x^s_{\text{tgt}}, y^s_{\text{tgt}})\}$ that share the same domain with y_{eng}. To bias the LLM further, we order the more relevant prompt examples closer to the input English example. Here, relevance between two examples is considered higher if they share the same intent. The remaining examples are

LLM INPUT:

Translate examples from English to Hindi

x^1_{eng} : Wake me up in two hours
y^1_{eng} : [IN:CREATE_ALARM [SL:DATE_TIME two hours]]
x^1_{tgt} : मुझे दो घंटे में जगाएं
y^1_{tgt} : [IN:CREATE_ALARM [SL:DATE_TIME दो घंटे]]

x^2_{eng} : Please set an alarm for 2 pm
y^2_{eng} : [IN:CREATE_ALARM [SL:DATE_TIME 2 pm]]
x^2_{tgt} : कृपया दोपहर दो बजे का अलार्म लगाएं
y^2_{tgt} : [IN:CREATE_ALARM [SL:DATE_TIME दो बजे]]

x_{eng} : Set the alarm for the flight next week
y_{eng} : [IN:CREATE_ALARM [SL:DATE_TIME next week]]

LLM OUTPUT:

x_{tgt} : फ्लाइट के लिए अगले सप्ताह अलार्म सेट करें
y_{tgt} : [IN:CREATE_ALARM [SL:DATE_TIME अगले सप्ताह]]

Figure 2: **Constructing the LLM Prompt** (§ 3.2): The input to the LLM contains a brief task description in the beginning followed by a series of English examples $(x^s_{\text{eng}}, y^s_{\text{eng}})$ and their translations in the target language $(x^s_{\text{tgt}}, y^s_{\text{tgt}})$ chosen from the seed sets $\mathcal{S}_{\text{eng}}$ and $\mathcal{S}_{\text{tgt}}$ respectively. Following the prompt examples, we append the new English example $(x_{\text{eng}}, y_{\text{eng}})$ to the input prompt which is fed to LLM. In the output, the LLM generates the translation for the new English example $(x_{\text{tgt}}, y_{\text{tgt}})$.

arbitrarily arranged to appear earlier in the prompt. Figure 2 shows an example translation—the *LLM input* contains two exemplars and then the English example that needs to be translated. The *LLM output* shows the translated output from the LLM.

3.3 Decoding Diverse Outputs from LLM

The text decoded from language models using the standard greedy decoding or beam search is often repetitive (Vijayakumar et al., 2016; Shao et al., 2017). To mimic how users express the same intentions in diverse ways, we experiment with the top-k and top-p sampling techniques (Fan et al., 2018; Holtzman et al., 2019) to decode multiple diverse translations per example. We expect sampling multiple translations to yield a better quality training dataset which in turn should result in better downstream semantic parsing performance compared to training on greedily decoded examples.

3.4 Data Filtering using Slot-Consistency

While the sampling techniques produce more diverse text, the sampled translations can be relatively noisy if they have lower likelihoods as per the model (Zhang et al., 2021). Thus, the translated pairs $(x_{\text{tgt}}, y_{\text{tgt}})$ in the LLM output can be

x_{eng} : Set the alarm for the flight next week
y_{eng} : [IN:CREATE_ALARM [SL:DATE_TIME next week]]

x^1_{tgt} : फ्लाइट के लिए अगले सप्ताह अलार्म सेट करें
y^1_{tgt} : [IN:CREATE_ALARM [SL:DATE_TIME अगले सप्ताह]]

x^2_{tgt} : अगले हफ्ते के लिए उड़ान के अलार्म को सेट करो
y^2_{tgt} : [IN:CREATE_ALARM [SL:DATE_TIME अगले हफ्ते के लिए]]

x^3_{tgt} : फ़्लाइट के लिए अलार्म सेट करें
y^3_{tgt} : [IN:CREATE_ALARM [SL:DATE_TIME अगले सप्ताह]]

x^4_{tgt} : अगली हफ्ते के उड़ान के लिए अलार्म सेट करें
y^4_{rgt} : [IN:CREATE_ALARM [SL:DATE_TIME अगले हफ्ते]]

Figure 3: **Slot Consistency Based Filtering** (§ 3.4): We present the input English example $(x_{\text{eng}}, y_{\text{eng}})$ and its four translated samples $\{(x^i_{\text{tgt}}, y^i_{\text{tgt}})\}$ the target language. The first two samples are *slot-consistent* as the slot-values (in green) in the logical forms appear exactly in the text utterances, while the last two samples are *slot-inconsistent* as the slot-values (in red) do not appear as an exact sub-string of the text utterance.

inconsistent w.r.t. each other. For example, consider the LLM translated pair $(x^3_{\text{tgt}}, y^3_{\text{tgt}})$ shown in Figure 3. Here, y^3_{tgt} contains a slot value (in red) that does not appear in the corresponding utterance x^3_{tgt} making the pair $(x^3_{\text{tgt}}, y^3_{\text{tgt}})$ inconsistent. As per the task definition, for a given example (x, y), the slot-values in the logical form y should come from the spans of the utterance x. Thus, we filter out the translated examples $(x_{\text{tgt}}, y_{\text{tgt}})$ like these where the slot-values in y_{tgt} do not appear *exactly* as an exact sub-span in x_{tgt}. Figure 3 shows examples of slot-consistent and slot-inconsistent generations by an LLM through top-k sampling.

4 Experimental Set-up

We describe our experimental setup in this section.

Datasets We experiment on two public datasets — MTOP (Li et al., 2021) and MASSIVE (FitzGerald et al., 2022). MTOP contains examples from six languages: English, French, German, Hindi, Spanish, and Thai, spanning 11 domains covering 117 intents and 78 slot types. On average, MTOP contains 12.3K examples in the train split, 1.5K in the dev split, and 2.7K in the test split per language. MASSIVE contains examples from 51 typologically diverse languages including English spanning 18 domains covering 60 intents and 50 slot types. For each language, MASSIVE contains roughly 11.5K examples in the train split, 2K examples in the dev split and 3K examples in the test split.

Evaluation Metric Prior work (Li et al., 2021; Nicosia et al., 2021) uses Exact Match (EM) accuracy as a primary metric which compares predicted and gold logical-forms strings. However, the exact string-match penalizes correct predictions where the order of slots within an intent is different. For example, consider the following logical-forms:

LF-1: `[IN:GET_WEATHER [SL:ATTRIBUTE rainfall] [SL:DATE today ] ]`

LF-2: `[IN:GET_WEATHER [SL:DATE today][SL:ATTRIBUTE rainfall ] ]`

LF-1 and LF-2 are equivalent but the difference in the ordering of slots results in a negative match. Thus, we correct the EM metric by making the match function agnostic to the ordering of slots within an intent in the logical-form. We compare different models as per this corrected EM metric.

Semantic Parsing Model We use a pre-trained mT5-Large checkpoint (1.2B parameters) to initialize the downstream semantic parsing models that map utterances in the input to logical-forms in the output. We finetune the mT5 model on the original English dataset mixed with the translated datasets in target languages. We train using the Adafactor optimizer (Shazeer and Stern, 2018) with a fixed learning rate of $1e{-}3$ and a batch size of 256, for 30K steps using the T5X library (Roberts et al., 2022) on 64 TPU-v3 chips. Examples from each language are sampled uniformly for batch creation. For model selection, we choose the best performing checkpoint as per the dev splits and report our results on the test splits.

LLM-T (Our Method) We experiment with 8B, 62B, and 540B sized variants of PaLM (Chowdhery et al., 2022) as our LLM, and primarily utilize LLM-540B for translating English examples in different languages. For the seed set $\mathcal{S}_{\text{tgt}}$ used for prompting the LLM, we borrow roughly 250 examples covering 11 domains from MTOP's train set and 350 examples covering 18 domains from MASSIVE's train set (§ 3.1). During decoding, we sample 8 translations per example using top-p sampling (§ 3.3), with $p = 0.95$ and temperature scaling $T = 0.7$, followed by filtering out slot-inconsistent examples (§ 3.4). We present an analysis of our design choices in § 5.3.

Baselines (i) **Zero-Shot**: Train the model only on the English data and evaluate on other languages in a zero-shot manner. (ii) **Few-Shot**: In addition to the English training data, use the seed set of examples $\mathcal{S}_{\text{tgt}}$ for each language during training. For MTOP, $|\mathcal{S}_{\text{tgt}}| \approx 250$ and for MASSIVE, $|\mathcal{S}_{\text{tgt}}| \approx 350$. (iii) **TAF**: We implement the method from Nicosia et al. (2021) that uses an off-the-shelf translation service (§ 2.2) to construct $\mathcal{D}_{\text{tgt}}$ in all the target languages. We borrow $\mathcal{D}_{\text{tgt}}$ from Nicosia et al. (2021) for MTOP and from Nicosia and Piccinno (2022) for MASSIVE.

5 Results and Analysis

We first present downstream performance of semantic parsing models trained on data generated by our method (§ 5.1) and compare with zero-shot setting, few-shot setting, and the TAF method (Nicosia et al., 2021). We then compare our method against the "full-shot" skyline where we utilize the original training datasets that were manually translated with the help of human annotators in the target languages (§ 5.2). We then present an analysis of different design choices that result in effective data translation using LLM-T (§ 5.3). Finally, we present an error analysis to show the key sources of disagreements between the parser predictions and the ground truth (§ 5.4). All the experiments use our corrected EM metric (§ 4; Evaluation Metric).

5.1 Evaluation on MTOP and MASSIVE

In Table 1, we compare performance of different methods for the 5 non-English languages in the MTOP dataset. The Zero-Shot baseline trains an mT5 model only on the English part of the train-split. The Few-Shot baseline additionally includes the human translated seed sets $\mathcal{S}_{\text{tgt}}$ for each language. Both TAF and LLM-T train on the original English train set mixed with their respective translated datasets in each language. As all the baselines utilize the original English train set, we see comparable performance on English (around 85.0 EM). We observe LLM-T outperforms TAF in 4 out of 5 languages by 3.6 EM. Since LLM-T uses $\mathcal{S}_{\text{tgt}}$ for prompting, we also mix $\mathcal{S}_{\text{tgt}}$ with TAF data and still observe that LLM-T improves over TAF+Few-Shot by 2.9 EM. On relatively low-resource languages, Hindi (`hi`) and Thai (`th`), LLM-T leads to much larger improvements over TAF.

Figure 4 shows the performance difference between our LLM-T method and TAF for the MASSIVE dataset (FitzGerald et al., 2022). On 41 out of 50 languages, we find LLM-T to be better than TAF. For nine languages LLM-T outperforms TAF by more than 5.0 EM—Simple Man-

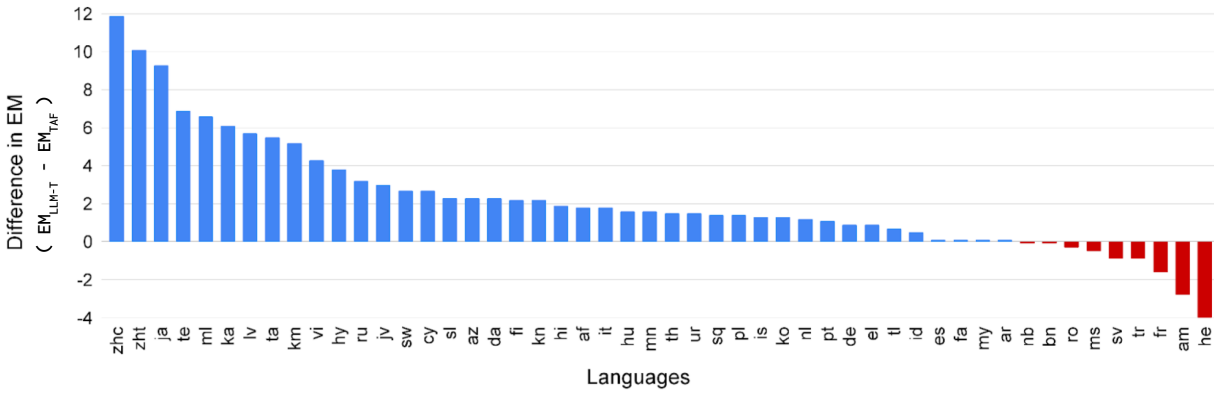

Figure 4: **EM accuracy difference between LLM-T and TAF across the 50 languages in MASSIVE dataset** (§ 5.1). LLM-T outperforms TAF on 41 out of 50 languages, with gains of more than 5 EM for nine of these languages. Only for Hebrew (he), LLM-T performs worse than TAF by more than 3 EM.

Method	de	es	fr	hi	th	Avg
Zero-Shot	54.4	57.8	62.8	42.3	42.1	51.9
Few-Shot	62.8	69.5	65.9	55.3	53.9	61.5
TAF	75.0	74.9	78.0	63.0	60.8	70.3
TAF + Few-Shot	**75.1**	74.5	78.5	63.9	62.9	71.0
LLM-T (ours)	74.0	**75.4**	**79.6**	**72.3**	**68.0**	**73.9**

Table 1: **EM accuracy comparison on MTOP** (§ 5.1): Data generated using LLM-T yields better performance on 4 out of 5 languages in MTOP. We observe large improvements for low-resource languages hi and th.

darin (zhc, +11.9), Traditional Mandarin (zht, +10.1), Japanese (ja, +9.3), Telugu (te, +6.9), Malayalam (ml, +6.6), Kannada (ka, +6.1), Latvian (lv, +5.7), Tamil (ta, +5.5), and Khmer (km, +5.2). Only for Hebrew (he, −4.0), LLM-T is worse by more than 3.0 EM. Averaged across all languages, LLM-T outperforms TAF by 2.2 EM. In Appendix A.1, we provide detailed baseline comparisons for all the 50 languages.

5.2 Comparison with gold translations

An ideal translate-train method should be competitive w.r.t. training on fully human translated datasets. Table 2 provides a comparison between training on TAF, LLM-T, and the datasets fully translated with the help of human annotators in the target languages (Gold). Between TAF and Gold, we observe a significant gap of 9.2 EM in MTOP and 6.7 EM in MASSIVE. Our method LLM-T, reduces this gap by 3.6 EM in MTOP and 2.2 EM in MASSIVE. Overall, LLM-T achieves roughly 93% of the performance obtained by the Gold skyline that use more than 30× human translated examples. Appendix A.1, provides per-language comparisons with the Gold skyline for both the datasets.

Dataset	Few-Shot	TAF	LLM-T	Gold
MTOP	61.5	70.3	73.9	**79.5**
MASSIVE	55.9	61.0	63.2	**67.7**

Table 2: **Comparison with Gold skyline** (§ 5.2): While training on the human translated datasets (Gold) yields the best performance, LLM-T results in a smaller performance gap compared to TAF. All numbers are averaged over the 5 non-English languages in MTOP.

Decoding Strategy	de	es	fr	hi	th	Avg
Greedy	71.1	71.7	72.6	68.1	66.0	69.9
+ Filtering	72.2	73.5	74.8	71.5	67.4	71.9
Top-p Sampling ($p = 0.95$)						
(#samples)						
1	70.1	71.5	74.3	66.9	67.2	70.0
2	71.4	72.1	74.5	68.8	67.2	70.8
4	71.1	72.8	76.4	69.0	66.0	71.1
8	71.9	72.7	74.2	70.0	**68.4**	71.4
Top-p Sampling + Filtering ($p = 0.95$)						
(#samples)						
1	72.0	75.2	78.9	71.6	68.1	73.2
2	73.7	75.2	79.5	72.0	67.6	73.6
4	73.4	75.3	79.0	72.1	67.7	73.5
8	**74.0**	**75.4**	**79.6**	**72.3**	68.0	**73.9**

Table 3: **Impact of decoding strategy and filtering**: Generating multiple translations per English example using top-p sampling followed by filtering inconsistent examples offers superior downstream performance compared to using greedy decoding or sampling just one translation per example. In Appendix A.2 we present results for top-k sampling as well.

Max Len	de	es	fr	hi	th	Avg
768	73.4	75.4	76.9	73.1	69.7	73.7
1024	74.0	75.4	79.6	72.3	68.0	73.9
1792	**74.3**	**75.7**	**80.5**	**74.0**	**71.1**	**75.1**

Table 4: **Impact of prompt length**: Longer prompts containing more exemplars result in more effective translated datasets yielding higher EM accuracy.

	de	es	fr	hi	th	Avg
LLM-T-8B	65.3	69.4	70.7	56.6	55.1	62.0
LLM-T-62B	72.0	73.3	76.7	68.2	65.6	71.2
LLM-T-540B	**74.0**	**75.4**	**79.6**	**72.3**	**68.0**	**73.9**

Table 5: **Impact of LLM size**: EM performance of semantic parsers trained on translated datasets improve with increasing the size of LLMs used for translation.

5.3 Analysis of Design Choices

We now present an analysis of the design choices that enabled more effective data translation via LLM-T. All the experiments in this section are carried out on the MTOP dataset.

Role of decoding strategy and filtering In Table 3, we present the EM accuracy of parsers trained on datasets translated using various combinations of decoding (§ 3.3) and filtering (§ 3.4) methods. For generating the translated outputs we experiment with greedy decoding, top-k (Fan et al., 2018) and top-p (Holtzman et al., 2019) sampling. Like prior translate-train methods, we begin with only one translation per example and observe sampling to be comparable with greedy decoding in downstream EM accuracy. In contrast, decoding two translations per example via sampling boosts the EM accuracy across all the languages. However, further increasing the translated samples to 4 and 8 results in only marginal performance differences. Manual inspection of the translated data revealed inconsistent utterance and logical-form pairs which motivated our design of slot-consistency based filtering (§ 3.4). Training the parser on filtered data provides further gains over training on unfiltered data. In Appendix A.2, we also present the results for top-k sampling. Overall, utilizing upto 8 top-p translated samples per English example followed by slot-consistency filtering provides the best performance averaged over all the languages.

Impact of Prompt Length We expect prompts containing more exemplars to yield higher quality translated examples owing to more information for in-context learning. In Table 4, we compare EM performance when using maximum prompt-lengths of 768, 1024, and 1792 tokens. Training on datasets translated using prompt-length of 1792 tokens provides the best downstream EM performance across all the languages. However, longer prompts lead to considerably longer inference times. Hence, we conduct our main experiments with prompt the length of 1024 tokens.

Role of LLM size In Table 5, we compare parser performance when trained on data generated by LLMs of different sizes. Training on larger LLM generated data leads to better performance—LLM-T-540B yields the best performance on all the languages, followed by LLM-T-62B which outperforms LLM-T-8B on all the languages.

5.4 Error Analysis

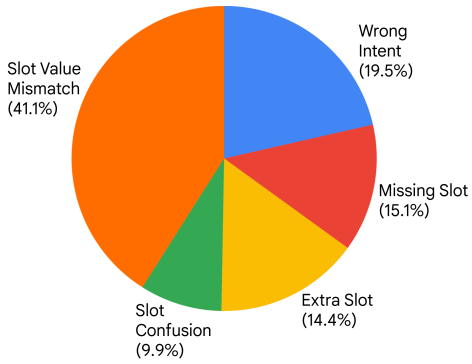

Figure 5: **Distribution of error categories**: estimated across all five languages on MTOP's dev set.

We analyze the examples where the predictions from our semantic parser do not match with the ground truth. In Table 6, we categorize all the erroneous examples into five broad categories (with English examples): (i) Slot Value Mismatch (ii) Wrong Intent (iii) Missing Slot (iv) Extra Slot and (v) Slot Confusion. Figure 5 presents the distribution of the error categories aggregated across all the languages on the MTOP dev-split. The "Slot Value Mismatch" is the most frequent error category (41.1%)—here the predicted parse structure is correct but the slot-values do not match perfectly with the gold parse. After manually inspecting 300 such errors we found that in roughly 50% of the cases the predicted and gold slot-values often have minor mismatches which may not be recognized as error by another human annotator and should not lead to incorrect output upon logical form execu-

Slot Value Mismatch (41.1%)	Utterance: Set an alarm for 5 pm tomorrow Prediction: [IN:CREATE_ALARM [SL:DATE_TIME for 5 pm] [SL:DATE_TIME tomorrow] Target: [IN:CREATE_ALARM [SL:DATE_TIME 5 pm] [SL:DATE_TIME tomorrow]
Wrong Intent (19.5%)	Utterance: What can I do today Prediction:[IN:QUESTION_NEWS [SL:DATE_TIME today]] Target: [IN:GET_EVENT [SL:DATE_TIME today]]
Missing Slot (15.1%)	Utterance: Play Justin Timberlake 's newest single Prediction:[IN:PLAY_MUSIC [SL:MUSIC_TYPE single]] Target: [IN:PLAY_MUSIC [SL:MUSIC_ARTIST_NAME Justin Timberlake] [SL:MUSIC_TYPE single]]
Extra Slot (14.4%)	Utterance: play music on the speaker Prediction: [IN:PLAY_MUSIC [SL:MUSIC_TYPE music] [SL:MUSIC_TYPE speaker]] Target: [IN:PLAY_MUSIC [SL:MUSIC_TYPE music]]
Slot Confusion (9.9%)	Utterance: audio call wedding planner please Prediction:[IN:CREATE_CALL [SL:CONTACT wedding planner]] Target: [IN:CREATE_CALL [SL:GROUP wedding planner]]

Table 6: **Examples of Error Categories** (§ 5.4) The errors in the predicted parse can be broadly classified into five categories: (i) Slot Value Mismatch: Predicted parse has the correct signature but the slot-values are incorrect, (ii) Wrong Intent: High-level intent of the predicted parse is incorrect, (iii) Missing Slot: One or more slots in the gold parse do not appear in the output, (iv) Extra Slot: Output contains extra slot(s) compared to the gold, (v) Slot Confusion: Prediced parse contains the correct correct intent and number of slots but the wrong slot-types.

tion. For example, in the first row of Table 6, the predicted value for the DATE_TIME slot is 'for 5 pm', while the target value is just '5 pm'.

6 Related Work

Multilingual Semantic Parsing Multilingual semantic parsers are typically initialized with a foundation model (Bommasani et al., 2021) pre-trained on vast amounts of multilingual data (Conneau et al., 2020; Xue et al., 2021; Li et al., 2021; FitzGerald et al., 2022) followed by supervised training on synthetic or real multilingual datasets. A standard approach for constructing multilingual datasets is to translate and localize English datasets with the help of multilingual speakers or machine translation. For example, MTOP (Li et al., 2021), MASSIVE (FitzGerald et al., 2022), and MultiAtis++ (Xu et al., 2020) were constructed by translating TOP (Gupta et al., 2018), SLURP (Roberts et al., 2022), and ATIS (Price, 1990) respectively through human translators.

Machine Translation based methods Machine translation based approaches continue to be important for multilingual task-specific models (Hartrumpf et al., 2008; Liang et al., 2020; Hu et al., 2020; Fang et al., 2021; Ladhak et al., 2020) including semantic parsing. Machine translation can either be used during the inference time to translate a user query into English for feeding it to an English-only model. This approach is referred to as *translate-test* (Artetxe et al., 2020; Uhrig et al., 2021). A more common way of using machine translation is in the form of data augmentation, referred as *translate-train* where English text in training data is translated into several languages (Sherborne et al., 2020; Moradshahi et al., 2020, 2021; Xia and Monti, 2021; Nicosia et al., 2021; Gritta et al., 2022; Wang et al., 2022). In practice, *translate-train* methods tend to outperform *translate-test* methods while also reducing the latency associated with translating text during the inference time (Yang et al., 2022).

LLMs and Few-Shot learning Transformer (Vaswani et al., 2017) based generative LLMs (Radford et al., 2019; Brown et al., 2020; Thoppilan et al., 2022; Soltan et al., 2022; Smith et al., 2022; Zhang et al., 2022; Chowdhery et al., 2022) trained on massive amounts of web-scale text corpora using next token prediction objective exhibit strong few-shot generalization abilities. When prompted with a task description and a handful of task-specific examples, LLMs can often match the performance of finetuned models via in-context learning (Xie et al., 2021; Min et al., 2022; Wei et al., 2022; Zhou et al., 2022). We utilize LLMs for translating English datasets in several languages using few-shot prompting.

7 Conclusion

We present a method of utilizing large language models (LLMs) for bootstrapping multilingual semantic parsers across several languages. In comparison to using off-the-shelf translation services that rely on significant amounts of human supervision, we demonstrate that prompting self-supervised LLMs can be a more effective and scalable alter-

native for dataset translation. We find that generating multiple diverse translations using sampling techniques followed by consistency-based filtering make the translated datasets more effective for training multilingual semantic parsers. On 41 out of 50 typologically diverse languages within two large datasets spanning several domains, our method outperforms a strong translate-train method that utilizes a supervised translation service.

8 Limitations

While translating English queries in different languages is a useful form of data augmentation, we think that further performance improvements can be obtained by careful localization of entities in the text queries. This will result in examples where the training dataset contains entities that are often talked about in the target language and might lead to less train-test domain shift. LLMs contain language specific priors which can be harnessed to perform such localization of the translated queries thus enabling more realistic data augmentations. In this work we presented a simple string-match based filtering technique to remove noisy translations. Data filtering can be further improved with the help of learned models. We observed that larger LLMs are important to generate more effective translated data. However running these experiments is constrained by the availability of large amounts of compute resources. We hope future work will address these limitations of our approach.

9 Ethical Considerations

We utilize large language models to translate datasets initially available in English into several languages. The real-world deployment of models trained on LLM-translated data should undergo a careful review of any harmful biases. However, the LLM-translated data and the logical-forms generated by a semantic parser are not user-facing, thus a smaller risk of any direct harms. The intended users of any semantic parsing model must be made aware that the answers returned by the model could be incorrect, more so for user-queries in low-resource languages. We do not immediately foresee any serious negative implications of the specific contributions that we make in this work.

10 Acknowledgment

We thank Rahul Goel, Slav Petrov, and Kartikeya Badola for discussions and their feedback on an earlier draft of this paper. We thank Massimo Nicosia for sharing the TAF translated datasets and helpful discussions during this project.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2020. Translation artifacts in cross-lingual transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7674–7684.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.

Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. 2021. Filter: An enhanced fusion method for cross-lingual language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12776–12784.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, et al. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *arXiv preprint arXiv:2204.08582*.

Milan Gritta, Ruoyu Hu, and Ignacio Iacobacci. 2022. CrossAligner & co: Zero-shot transfer methods for task-oriented cross-lingual natural language understanding. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4048–4061, Dublin, Ireland. Association for Computational Linguistics.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792.

Sven Hartrumpf, Ingo Glöckner, and Johannes Leveling. 2008. Efficient question answering with question decomposition and multiple answer streams. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 421–428. Springer.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Sawan Kumar and Partha Talukdar. 2021. Reordering examples helps during priming-based few-shot learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4507–4518.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online. Association for Computational Linguistics.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Mehrad Moradshahi, Giovanni Campagna, Sina Semnani, Silei Xu, and Monica Lam. 2020. Localizing open-ontology QA semantic parsers in a day using machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5970–5983, Online. Association for Computational Linguistics.

Mehrad Moradshahi, Victoria Tsai, Giovanni Campagna, and Monica S Lam. 2021. Contextual semantic parsing for multilingual task-oriented dialogues. *arXiv preprint arXiv:2111.02574*.

Massimo Nicosia and Francesco Piccinno. 2022. Evaluating byte and wordpiece level models for massively multilingual semantic parsing. *arXiv preprint arXiv:2212.07223*.

Massimo Nicosia, Zhongdi Qu, and Yasemin Altun. 2021. Translate & fill: Improving zero-shot multilingual semantic parsing with synthetic data. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3272–3284.

P. J. Price. 1990. Evaluation of spoken language systems: the ATIS domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, et al. 2022. Scaling up models and data with `t5x` and `seqio`. *arXiv preprint arXiv:2203.17189*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2210–2219.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Tom Sherborne, Yumo Xu, and Mirella Lapata. 2020. Bootstrapping a crosslingual semantic parser. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 499–517.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, et al. 2022. Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model. *arXiv preprint arXiv:2208.01448*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

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.

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.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Zhiruo Wang, Grace Cuenca, Shuyan Zhou, Frank F Xu, and Graham Neubig. 2022. Mconala: A benchmark for code generation from multiple natural languages. *arXiv preprint arXiv:2203.08388*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Menglin Xia and Emilio Monti. 2021. Multilingual neural semantic parsing for low-resourced languages. In *The Tenth Joint Conference on Lexical and Computational Semantics*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual nlu. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5052–5063.

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, Online. Association for Computational Linguistics.

Diyi Yang, Ankur Parikh, and Colin Raffel. 2022. Learning with limited text data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 28–31, Dublin, Ireland. Association for Computational Linguistics.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *AAAI/IAAI*, pages 1050–1055, Portland, OR. AAAI Press/MIT Press.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, page 658–666, Arlington, Virginia, USA. AUAI Press.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Appendix

A.1 Additional results

Lang	Zero-Shot	Few-Shot	TAF	TAF + Few-Shot	LLM-T (top-k)	LLM-T (top-p)	Gold (skyline)
af	48.5	59.0	64.5	64.5	66.7	66.3	68.5
am	31.0	47.6	58.3	57.4	56.1	55.5	64.6
ar	35.9	50.5	57.2	58.0	56.6	57.3	65.5
az	39.3	57.1	60.5	60.5	62.6	62.8	68.8
bn	40.8	55.4	62.1	61.7	61.1	62.0	68.3
cy	26.7	44.8	59.1	58.3	61.1	61.8	65.1
da	57.5	62.4	66.2	66.3	69.1	68.5	71.0
de	54.3	62.8	67.5	67.7	68.3	68.4	70.4
el	47.3	57.8	64.2	65.5	65.2	65.1	68.7
en	72.7	71.4	73.5	72.9	73.3	73.4	73.0
es	53.4	58.1	64.6	64.6	64.7	64.7	66.6
fa	48.8	58.0	63.1	62.9	62.8	63.2	68.1
fi	47.5	58.4	65.0	65.3	66.7	67.2	70.9
fr	54.6	58.0	65.3	64.9	63.9	63.7	67.1
he	35.3	56.1	60.6	61.2	55.3	56.6	68.3
hi	40.1	54.4	61.6	62.5	63.1	63.5	66.2
hu	44.1	57.1	63.8	63.6	64.5	65.4	69.7
hy	39.3	53.8	58.7	59.2	62.3	62.5	67.1
id	55.3	60.2	65.5	65.9	66.6	66.0	69.1
is	41.3	54.4	62.2	61.5	63.6	63.5	69.5
it	52.3	58.6	64.0	63.6	65.2	65.8	67.2
ja	45.6	55.1	56.3	56.5	65.6	65.6	67.3
jv	34.3	51.7	58.6	60.2	62.0	61.6	66.7
ka	36.5	53.4	53.5	54.6	59.2	59.6	65.7
km	37.8	51.1	49.1	53.7	55.3	54.3	62.8
kn	37.1	49.3	55.0	55.9	57.7	57.2	62.1
ko	42.1	56.3	62.2	63.6	62.4	63.5	69.3
lv	45.4	56.0	60.4	61.3	66.0	66.1	68.8
ml	38.6	53.9	55.5	56.9	62.5	62.1	67.5
mn	30.9	51.4	57.6	59.4	59.5	59.2	68.0
ms	48.6	58.9	66.2	66.2	65.8	65.7	69.2
my	38.1	54.9	60.5	62.3	61.5	60.6	69.6
nb	55.2	63.0	67.5	67.7	67.7	67.4	71.0
nl	53.1	61.2	67.3	68.5	68.7	68.5	70.5
pl	50.5	57.4	61.1	61.4	62.9	62.5	65.6
pt	54.9	60.3	65.8	65.7	66.4	66.9	68.5
ro	51.2	58.8	65.4	65.0	64.8	65.1	68.8
ru	42.3	59.4	63.0	63.1	66.6	66.2	69.4
sl	46.0	57.8	63.1	64.0	65.3	65.4	68.8
sq	41.0	55.4	60.3	60.4	62.1	61.7	67.3
sv	57.2	63.1	69.8	69.6	69.3	68.9	72.4
sw	35.7	52.3	57.9	57.5	60.9	60.6	65.3
ta	37.2	53.0	55.4	55.7	60.7	60.9	65.8
te	38.7	49.0	51.6	53.6	56.8	58.5	61.6
th	49.4	60.0	63.5	66.5	65.2	65	71.5
tl	48.4	55.7	64.1	64.2	65.2	64.8	67.5
tr	46.7	58.5	63.7	63.4	62.7	62.8	69.4
ur	38.9	51.2	60.4	60.6	62.2	61.9	64.6
vi	46.9	55.1	59.0	59.2	63.0	63.3	67.6
zhc	34.7	56.1	52.0	53.9	64.2	63.9	66.3
zht	35.2	51.8	50.5	52.3	60.7	60.6	63.6
Avg	43.8	55.9	61.0	61.6	63.2	63.2	67.7

Table A1: EM accuracy comparison on MASSIVE dataset. Avg reports the EM accuracy averaged across the 50 non-English languages

Lang	Zero-Shot	Few-Shot	TAF	TAF + Few-Shot	LLM-T (top-k)	LLM-T (top-p)	Gold (skyline)
de	54.4	62.8	75.0	75.1	73.7	74.0	78.5
es	57.8	69.5	74.9	74.5	75.2	75.4	82.9
fr	62.8	65.9	78.0	78.5	79.7	79.6	80.8
hi	42.3	55.3	63.0	63.9	72.5	72.3	78.5
th	42.1	53.9	60.8	62.9	66.8	68.0	77.0
en	84.1	84.0	85.2	85.0	85.2	85.1	85.4
Avg	51.9	61.5	70.3	71.0	73.6	73.9	79.5

Table A2: EM accuracy comparison on MTOP dataset. Avg reports the EM accuracy averaged across the 5 non-English languages

In Table A1, we present detailed baseline comparisons for all the 51 languages in the MASSIVE dataset. Zero-Shot, Few-Shot, TAF, and TAF+Few-Shot are the baselines described in Section 4. LLM-T represents our method with top-k or top-p sampling used while decoding the translated examples. Gold is the "full-shot" skyline which utilizes the original human-translated datasets (§ 5.2). Table A2 presents the same set of results for the six languages in the MTOP dataset.

A.2 Role of decoding strategy and filtering

In Table A3 we present results for different decoding strategies and role of filtering inconsistent examples as discussed in Section 5.3.

Decoding Strategy	de	es	fr	hi	th	Avg
Greedy	71.1	71.7	72.6	68.1	66.0	69.9
+ Filtering	72.2	73.5	74.8	71.5	67.4	71.9
Top-k Sampling ($k = 40$)						
(#samples)						
1	70.5	71.7	73.1	66.8	66.5	69.6
2	72.3	72.7	75.7	68.7	67.3	71.3
4	71.3	73.1	73.8	68.5	67.8	70.9
8	71.1	72.5	74.2	69.3	67.5	70.9
Top-k Sampling + Filtering ($k = 40$)						
(#samples)						
1	72.4	74.4	78	70.9	66.1	72.4
2	73.6	74.4	78.2	72.1	67.9	73.2
4	73.4	75.3	78.8	71.4	67.1	73.2
8	73.7	75.2	**79.7**	**72.5**	66.8	73.6
Top-p Sampling ($p = 0.95$)						
(#samples)						
1	70.1	71.5	74.3	66.9	67.2	70.0
2	71.4	72.1	74.5	68.8	67.2	70.8
4	71.1	72.8	76.4	69.0	66.0	71.1
8	71.9	72.7	74.2	70.0	68.4	71.4
Top-p Sampling + Filtering ($p = 0.95$)						
(#samples)						
1	72.0	75.2	78.9	71.6	**68.1**	73.2
2	73.7	75.2	79.5	72.0	67.6	73.6
4	73.4	75.3	79.0	72.1	67.7	73.5
8	**74.0**	**75.4**	79.6	72.3	68.0	**73.9**

Table A3: **Impact of decoding strategy and filtering**: Generating multiple translations per English example using top-k or top-p sampling followed by filtering inconsistent examples offers superior downstream performance compared to using greedy decoding or sampling just one translation per example.