Localizing Q&A Semantic Parsers for Any Language In a Day

Mehrad Moradshahi Giovanni Campagna Sina J. Semnani Silei Xu Monica S. Lam
Computer Science Department
Stanford University
Stanford, CA, USA
{mehrad,gcampagn,sinaj,silei,lam}@cs.stanford.edu

Abstract

We propose Semantic Parser Localizer (SPL), a toolkit that leverages Neural Machine Translation (NMT) systems to localize a semantic parser for a new language. Our methodology allows for automatic generation of training data in the target language by augmenting machine translated datasets with local property values scraped from public websites, trains a semantic parser for the target languages, then validates and tests the model on natural utterances curated using human translators.

We assess the effectiveness of our approach by extending the current capabilities of a recently-proposed system for Question Answering (QA) on the open web to 10 foreign languages for the restaurants and hotels domains. We show that best performance can be achieved using a few shot approach where a small proportion of the train set consists of natural human translations of utterances from the English development set. Our model achieves an overall test accuracy ranging between 64.41% and 79.12% for the hotels domain and between 75.6% and 82.4% for restaurants domain, which compares favorably to the English parser trained on gold English data. Our methodology enables any software developer to add a new language capability to any QA system for a new domain in less than 24 hours.[1]

1 Introduction

Localization is an important step in software or website engineering as it allows the product to reach an international audience in their native language. It is well supported with professional services that can translate text strings quickly into a wide variety of languages. As conversational agents are increasingly used as the new interface, how do we localize them to other languages?

The focus of this paper is on question answering systems that use *semantic parsing*, where natural language is translated into a formal, executable representation (such as SQL). Semantic parsing typically requires a large amount of training data, which must be annotated by an expert of the formal language. It is difficult to find experts that understand all the different languages needed for localization. For English, previous work has shown it is possible to bootstrap a semantic parser without massive amount of manual annotation, by using a large, hand-curated grammar of natural language (Wang et al., 2015; Xu et al., 2020). This approach is still too expensive to replicate for all languages, due to the effort and expertise required to build such a grammar. Hence, we investigate the question: *Can we leverage previous work on English semantic parsers for other languages?* And in particular, can we do so without requiring experts in each language?

We propose adding a multi-lingual capability directly into a synthesis-based parser generation toolkit, such as Schema2QA, so we can easily create parsers for new domains in different languages. The challenge is that while neural machine translation (NMT) technology can handle the general structure of the question well, the parser must be able to answer questions about locale-specific real-world entities, such as hotel chains, restaurant names, and local cuisines.

We propose a methodology that requires no manual annotations of the target language at all. We first translate the English training set into the target language using NMT, while keeping the parameter values in the sentences intact. We then replace the translated sentences with new native parameter entities in the target language to create training and evaluation data in the target language, with local-

[1] We will be releasing our datasets and pretrained models to encourage further research in this area upon publication.

ISO code	Language	Domain
		Hotels
en	English	give me a list of all hotels with breakfast that have over 7 ratings .
ar	Arabic	'aetiny qayimat bijimye alfanadiq dhat wajibat 'iiftar walty biha tqyym 'aelaa min 7 .
de	German	geben sie mir eine liste aller hotels mit einem Frühstück , die über 7 bewertung haben .
es	Spanish	dame una lista de todos los hoteles con desayuno que tengan más de 7 calificaciones .
fa	Persian	bh mn lysty az htlhay nzdyk ra nshan bdh kh daray sbhanh ba hdaql amtyaz 7 bashnd .
fi	Finnish	anna minulle lista kaikista hotelleista joilla on aamiainen ja yli 7 arviota .
it	Italian	dammi una lista di tutti gli hotel con colazione con più di 7 recensioni .
ja	Japanese	7件以上のレビューがあるの朝ごはんのあるホテルのリストを教えてください。
pl	Polish	pokaż listę wszystkich hoteli wyposażonych w śniadanie z ponad 7 ocenami .
tr	Turkish	bana 7 üzeri puanlı kahvaltı içeren tüm otellerin listesini verir misin .
zh	Chinese	给我列出所有有7个以上评分的有早餐的酒店。
		Restaurants
en	English	search for italian restaurants with at least 3 reviews .
ar	Arabic	'abhath ean almataeim 'iitaliin mae 3 min altaealyqat ela alaql .
de	German	suche nach italienisch restaurants mit mindestens 3 bewertungen .
es	Spanish	buscar un restaurante italiano en le louverot con , al menos, 3 opiniones .
fa	Persian	rstwranhay aytalyayy ba hdaql 3 nqd w brrsy ra jstjw knyd .
fi	Finnish	etsi italialainen ravintoloita joilla on vähintään 3 arvostelua .
it	Italian	cerca ristoranti con cucina italiana con almeno 3 recensioni .
ja	Japanese	3件以上のレビューがあるのイタリアンレストランを検索してください。
pl	Polish	szukaj restauracji Włoski minimalną liczbą opinii: 3 .
tr	Turkish	en az 3 değerlendirmeye sahip italyan restoranları ara .
zh	Chinese	搜索有至少3个评价的意大利餐厅。

Table 1: Example of queries from schema2QA in English and 10 other languages. Sentences in each domain are semantically-equivalent and will be mapped to the same logical forms ignoring the parameter values. Arabic and Persian are right to left languages. We show their transliteration in this table for simplicity of presentation.

ized parameters. Additionally, a small sample of the synthesized English questions are translated by native speakers with no technical expertise, as a few-shot boost to the automatic training set and as test data.

We apply our approach on the *Restaurants* and *Hotels* datasets introduced by Xu et al. (2020), which contains complex queries on data scraped from major websites. We demonstrate the efficiency of our methodology by creating neural semantic parsers in Arabic, German, Spanish, Persian, Finnish, Italian, Japanese, Polish, Turkish, Chinese. The models can answer complex questions about hotels and restaurants in the respective languages. An example of a query is shown for each language and domain in Table 1. We collected a test set of professionally translated questions in each language from the existing English test sets.

Our contributions in this paper include the following:

- To the best of our knowledge, this is the first multi-lingual semantic parsing dataset with localized entities. Our dataset covers 10 linguistically different languages, to support a wide range of syntax. We hope that releasing our dataset will trigger further work and research in multilingual semantic parsing.
- Semantic Parser Localizer (SPL), a toolkit to localize a multilingual semantic parser for any language. With our proposed methodology, a semantic parser can be created for a new target language by collecting entities for the new language, and manually translating a couple of hundred sentences. No manual annotation of sentences is necessary.
- An improved neural semantic parsing model, based on BERT-LSTM (Xu et al., 2020) but using the XLM-R encoder. Although in this work, we have applied our model to a multilingual semantic parsing task, it can be deployed in multitask and cross-lingual learning settings for any NLP task that can be framed as question answering. We will be releasing our code and pre-trained models upon publication.
- Experimental results for our approach applied to answer questions on hotels and restaurants in 10 different languages. SPL achieves a logical form accuracy of 71.5% for hotels and 78.7% for restaurants averaged across all languages, which is comparable to the English parser trained with English synthetic and paraphrased data. Our method outperforms the previous state of the art and 3 strong baselines, by between 20% and 40%, depending on the language and domain.

2 Related Work

Multi-lingual benchmarks Previous work has shown it is possible to ask non-experts to annotate large datasets for applications such as natural language inference (Conneau et al., 2018) and machine reading (Clark et al., 2020), which has led to large cross-lingual benchmarks (Hu et al., 2020). Their approach is not suitable for semantic parsing, because it requires experts that know both the formal language and the natural language.

Semantic Parsing Previous work on semantic parsing is abundant, with work dating back to the 70s (Woods, 1977; Zelle and Mooney, 1996; Kate et al., 2005; Berant et al., 2013). State of the art methods, based on sequence-to-sequence neural networks, require large amounts of manually annotated data (Dong and Lapata, 2016; Jia and Liang, 2016). Various methods have been proposed to eliminate manually annotated data for new domains, using synthesis (Wang et al., 2015; Shah et al., 2018; Campagna et al., 2019; Xu et al., 2020), transfer learning (Zhong et al., 2017; Herzig and Berant, 2018; Yu et al., 2018), or a combination of both (Rastogi et al., 2019; Campagna et al., 2020). Nevertheless, these works focus mainly on the English language, and extending the current capabilities of such systems to new languages is still unsolved.

Cross-lingual Transfer of Semantic Parsing Duong et al. (2017) investigates cross-lingual transferability of knowledge from a source language to the target language by employing cross-lingual word embedding. They evaluate their approach on the English and German splits of NLmaps dataset (Haas and Riezler, 2016) and on a code-switching test set that combines English and German words in the same utterance. However, they found that joint training on English and German training data achieve competitive results compared to training multiple encoders and predicting logical form using a shared decoder. This calls for better training strategies and better use of knowledge the model can potentially learn from each example.

The closest concurrent work to ours is Bootstrap (Sherborne et al., 2020) where they explore using public MT systems to generate training data for other languages. They try different training strategies and find that using a shared encoder and training on target language sentences and unmodified logical forms with English entities yields the best result. Their evaluation is done on the ATIS (Dahl et al., 1994) and Overnight (Wang et al., 2015) datasets, in German and Chinese. These two benchmarks have a very small number of entities. Their method is unsuitable to applications with a large number of entities, as the semantic parsing model must jointly learn to identify the entity and translate it to English. This is very challenging in a setting with limited data.

To collect real validation and test utterances, Sherborne et al. (2020) use a three-staged process to collect data from Amazon Mechanical Turkers (AMTs). They ask for three translations each per English source sentence with the hypothesis that this will collect at least one adequate translation. However, we found this approach to be very time consuming and have less quality compared to using professional translators. Since this process is done once per language per data split, it's important for the translations to be verified and have high quality.

3 Multi-Lingual Parser Generation

Our goal is to add multilingual capability to a semantic parser generation toolkit, such as Schema2QA, so we can leverage new domains or new capabilities added to the system. In the following sections, we first describe Schema2QA, then how we introduce the multi-lingual capability into the system.

3.1 Synthesis-Based Semantic Parsers

Let us first review the manual steps in a semantic parser generator like Schema2QA (Xu et al., 2020).

1. Hand-curate generic question templates for the target language. English has about 600 generic question templates, covering different who, what, when, where questions, which can refer to an attribute with different parts-of-speech (POS) qualifiers such as "a restaurant that serves Italian food", or "an Italian restaurant".
2. Add manual annotations for each attribute in different Parts of Speech (POS).
3. Crowdsource workers paraphrase a portion of the sentences synthesized from the generic templates and the annotations.
4. Collect entity libraries relevant in each domain, which are then used to augment the synthesized and paraphrased sentences to create a training data set. Schema2QA proposes using existing

Schema.org metadata to automatically collect these from web crawling.

5. Collect a validation set of realistic questions, which are hand-annotated by an expert. The validation set is used to iterate the generic templates and the annotations.

3.2 Our Proposed Methodology

Our goal in designing the multi-lingual pipeline is (1) to leverage previous and future effort in creating English parsers as much as possible, (2) to not require annotating questions in the new language because such expertise is hard to find, and (3) to use human translators as minimally as possible.

Our proposed methodology skips the first three manual steps in the Schema2QA methodology described above. We need to, however, perform step 4 and replace step 5 with manual translation. A multilingual system intended to be used internationally must learn the field type using the semantics of sentences with local entities as parameters. We observe that schema.org is a standard worldwide, so international websites all use the same English schema properties, even though the contents are in the native language. We can scrape sites like Yelp and Wikipedia which have content in many languages. Instead of annotating the validation and test sets, we use professional translation services to translate the validation and test set splits of synthesized English sentences.

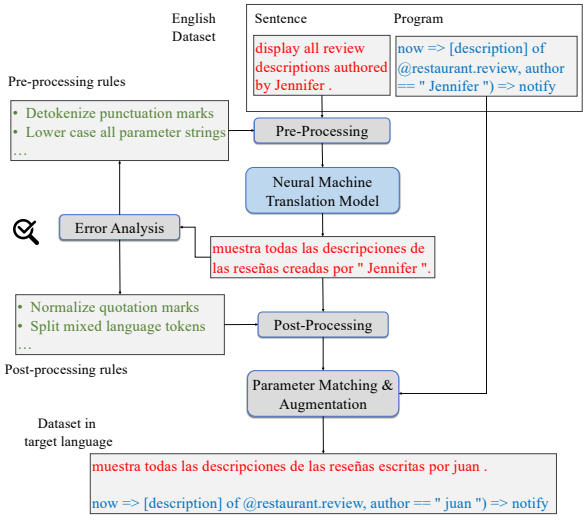

Figure 1: Dataset collection method used to generate translated train and validation splits. We obtain the test splits using human translation.

3.3 Data Set Generation Overview

Here we describe how we generate a large training dataset in a new language using local entity values from the training data created by Schema2QA. We do not use the paraphrase data because paraphrasing in English may introduce annotation errors, and machine translation automatically provides the same benefit as human paraphrasing. The public neural translation models tend to generate natural-looking sentences, particularly for popular language pairs.

Schema2QA's synthesized and augmented sentences contain many sentences that are identical if parameters are ignored. We first deduplicate them so there is only one sample for each sentence structure to reduce translation cost. Next, we translate the English sentences with English entity parameters into the new language, while keeping the English parameters intact. This ensures that the translated sentence matches the annotation. Finally, we augment the translated sentences by replacing the English parameters in the sentence and its program annotation with the real parameters in the target language. Figure 1 shows an example of this process.

3.4 Translation

This project leverages public NMT models such as Google Translate, which have been optimized to produce the correct translation for many different pairs of languages. Because we do not have access to their internal models, we cannot apply finetuning (Qiu et al., 2020) to adapt the translators to the semantic distribution of our inputs. However, from observing typical errors of a sample of inputs, we derive a few simple preprocessing and postprocessing rules to improve the NMT results for our purpose. For example, we found that tokenizing punctuation marks by inserting a space character between them and the last word in the sentence can yield different results when translating to Persian. Furthermore, for languages such as Chinese and Japanese where there is no whitespace delimitation between words in the sentence, the quotation marks are sometimes omitted during translation which makes parameter matching step difficult. We post-process the sentence using regular expressions to split English parameters from Chinese tokens.

It is important that the translator recognizes the parameters and keeps the parameters intact. Translation, by nature, is an invasive text processing

	Dataset				
Metrics	**Overnight (Blocks)**	**Overnight (Social)**	**ATIS**	**Schema2QA (Restaurants)**	**Schema2QA (Hotels)**
# examples	1,305	2,842	4,433	39,568	457,729
avg # unique unigrams per example	6.82	8.65	7.75	11.74	11.10
avg # unique bigrams per example	7.44	8.68	6.99	11.21	10.45
avg # unique trigrams per example	6.70	7.90	6.03	10.24	9.46
avg # properties per example	1.94	1.65	2.56	1.89	1.95
avg # values per property	≤ 2	≤ 2	≤ 20	≥ 100	≥ 100

Table 2: Statistical analysis of the training set for Overnight, ATIS, and schema2QA datasets. For overnight, the two domains with the lowest reported accuracies are chosen.

method where tokens can be modified, transliterated, omitted, or get mapped to a new token in the target language. If the semantics of the generated utterance in the target language is changed, the original program will no longer be its annotation.

In our approach, we first identify all the entity and property values in the input sentences, then we build a custom terminology which will be used during translation to mask the tokens specified in the glossary. This is especially effective for languages that change the shape of input words based on sentence context such as Arabic, Finnish, and Turkish.

Manually translated sentences are collected by asking each translator to provide the most natural written form of each sentence in their language, equivalent to how they would type their queries for a text-based virtual assistant. We show to the translators English sentences, where the actual parameters are either fully capitalized if they are named entities or wrapped in quotation marks for all other types of parameters such as time, duration, phone number, and postal codes. We ask the translators to keep the parameters intact and not translate them. The parameters in the sentences and their annotations are substituted later with local values in the target language.

4 Model Description

Our neural semantic parser is based on the previously proposed BERT-LSTM architecture (Xu et al., 2020), which we modify to use the XLM-R pretrained model (Conneau et al., 2019) as the encoder. The model is an encoder-decoder (Sutskever et al., 2014) neural network that uses a Transformer encoder (Vaswani et al., 2017) and a LSTM decoder with attention and pointer-generator (See et al., 2017). More details are provided in the supplementary document.

We apply rule-based preprocessing to identify numbers, times, dates, etc. All other tokens are lower cases and split into subwords according to the pretrained vocabulary. The same subword preprocessing is applied to entity names that are present in the output logical form.

5 Experiments

We have implemented the full SPL methodology in the form of a tool. Developers can use the SPL tool to translate any semantic parser for their task. In this section, we evaluate how well the semantic parsers created by our tool perform. We evaluate on a new challenging dataset for multilingual semantic parsing. We first describe our dataset, and then show the accuracy of our tool, both without any human-produced training data (zero-shot), and if human-created data in the target language is available (few-shot).

In our experiments, we measure the *logical form exact match* (em) accuracy, which considers the result to be correct only if the output matches the gold logical form token by token. We additionally measure the *structure match* (sm) accuracy, which measures whether the gold and predicted logical forms are identical, ignoring the parameter values. We report results on both validation and test sets. We present the results for both restaurants and hotels domain in this paper. Due to space constraints, we only provide the discussion on hotels' results. The hotels data set is more challenging, and the trends of both domains are similar.

Our tool was implemented using the Genie toolkit (Campagna et al., 2019) for synthesis and data augmentation. Our model was implemented using the Huggingface library (Wolf et al., 2019). Our code will be released open-source.

5.1 Dataset

Using our approach, we have constructed a multilingual dataset based on the previously proposed Schema2QA *Restaurants* and *Hotels* (Xu et al., 2020). We chose these two datasets as a starting point as require understanding both complex questions and a large number of entities, many of which

are not seen in training. Note that annotation of logical forms is *aligned* with the input utterance: every open-ontology parameter value must appear exactly in the utterance. Table 2 shows the comparison with the previously released Overnight (Wang et al., 2015) and ATIS datasets (Dahl et al., 1994), which previous work has translated to other languages (Sherborne et al., 2020). The Schema2QA dataset is larger, has more linguistic variety, and significantly more possible values for each property.

We have translated the Schema2QA dataset in 10 different languages, chosen to be linguistically diverse. The training set is generated by using public NMT systems. The validation and test sets are professionally translated. Hotels domain contains 331 and 340 examples, and restaurants domain contains 378 and 415 examples in the validation and test splits respectively. We postprocess all the sets and replace the parameters with local values according to their property types. We use uniform-sampling for train set and random-sampling for validation and tests. The dataset will be released upon publication.

5.2 BackTranslation: Translate at Test Time

As a baseline, we train a neural semantic parser on the English training set; at test time, the sentence is translated on-the-fly while leaving their parameter values intact, from the target language to English, and passed to the semantic parser.

The experimental results are shown in Table 3; 5 representatives of the 10 languages are presented here for lack of space. The hotel results from the BackTranslation baseline vary from a minimum of 31.18% for Persian to a maximum of 48.32% for Chinese. Comparing the results to English, we observe about 20% drop in exact match accuracy. This is reasonable considering that the English semantic parser has been validated and tested on real human utterances. We observe that BackTranslation produces natural-looking sentences with variety; in fact, back translation (English $\rightarrow$ Chinese $\rightarrow$ English) has been used in previous literature for generating paraphrases (Federmann et al., 2019).

The difference between structure match accuracy vs exact match accuracy is about 9%, and it is consistent across languages including English. This suggests there is a set of examples that the parser is predicting the wrong parameter values for independent of the language. This is mainly caused in two cases: First, the parameter value is ambiguous and can be used for two different property types. For example the word “California Pizza Kitchen" to a parser that has been trained with restaurant domain data is a *food place*. Whereas for a parser trained on out of domain data is a Kitchen *place* in California *state* which serves Pizza *cuisine*. Second, if the sentence is short, there is not enough context around the parameter values to inform the model of what the query is asking about. For example, the sentence “search for italian" can be understood as a place that serves italian cuisine or a food place named italian.

5.3 Bootstrap: Train with Translated Data

As proposed by Sherborne et al. (2020), we create a new training set by using NMT to translate the English sentences into the target language; the logical forms are left unmodified.

Bootstrap achieves between 14% to 25% accuracy which is significantly worse than BackTranslation results for all 6 languages. The reason for this drop is twofold: First, as parameter values are not preserved during translation, they can either get translated, transliterated, or mapped to new token(s) depending on the context. Furthermore, the mapping is not done deterministically since public APIs we use for translation are constantly updated and refined. One can potentially form a dictionary by finding all possible translations for each token in the ontology. This is not a feasible approach simply due to having an open ontology which can get updated over time. Second, since we are not able to identify the original parameters in the translated sentence using this approach, augmentation with real parameters from the native library is not possible. This step is much needed for the neural model to be able to generalize beyond the fixed set of values it has seen during training.

5.4 SPL: Our Semantic Parser Localizer

We generate the training set by applying our methodology on the training English dataset. We then train one semantic parser for each language and validate and test on human translated dataset.

The results obtained by applying our methodology outperforms all the previous baselines for all the languages we did experiments on. Specifically, we achieve improvements between 17% to 33% over the BackTranslation results. A neural semantic parser trained with Bootstrap approach must not only learn to identify entities and how their

	BackTranslation		Bootstrap				Bootstrap (+English)			
Language	**Test**		**Dev**		**Test**		**Dev**		**Test**	
	em	sm	em	sm	em	sm	em	sm	em	sm
Arabic	**43.82**	**52.06**	16.62	35.35	16.76	37.65	18.13	37.76	17.65	37.65
German	**38.53**	**49.12**	24.17	39.27	22.35	42.35	34.14	48.94	32.35	50.59
Persian	**31.18**	**40.88**	18.13	36.56	14.12	34.71	17.22	34.74	13.53	33.82
Japanese	**40.88**	**48.23**	25.08	44.41	25.88	44.71	26.28	48.64	25.29	45.29
Turkish	**33.53**	**44.71**	27.79	37.46	25.59	40.88	38.97	48.04	31.47	45.29
Arabic	**36.14**	**42.41**	16.93	39.68	9.88	36.14	18.25	47.62	11.81	40.72
German	**35.42**	**40.24**	30.42	54.23	30.12	52.53	30.69	49.21	29.16	52.29
Persian	**41.93**	**50.36**	17.46	30.42	12.77	31.81	17.2	37.04	12.29	35.42
Japanese	**46.99**	**56.14**	15.87	43.39	12.77	39.76	16.93	48.68	13.01	50.12
Turkish	**42.65**	**48.67**	23.81	35.71	17.35	34.7	24.6	41.27	17.59	39.04

Table 3: Experiment results for hotels (top rows) and restaurants (bottom rows) domain using Bootstrap and BackTranslation methods. em and sm indicate exact and structure match accuracy respectively. BackTranslation results for all languages and domains are provided in the supplementary material. Exact match accuracies for the English Test set are 69% for hotels, and 74% for restaurants.

Language	BLEU score	TER score
Arabic	0.23	0.59
German	0.32	0.53
Spanish	0.39	0.44
Persian	0.12	0.80
Finnish	0.22	0.57
Italian	0.29	0.53
Japanese	0.35	0.68
Polish	0.26	0.57
Turkish	0.18	0.64
Chinese	0.29	0.55

Table 4: Results for different metrics used to calculate the similarity between human translation and machine translation sentences for hotels evaluation set. Higher BLUE score and lower TER score means higher translation quality.

relationship maps to database operators, it must also translate (or transliterate) the entity names to English. The SPL neural model instead takes advantage of entity alignment in the utterance and logical form, and can copy the entity directly.

5.5 SPL (MT): Machine-Translated Validation Data

To understand the significance of using human translated data for valuation, here we evaluate our models on the Machine Translated (MT) development set. The difference between the results varies by language. The hand-translated validation set delivers better results on 7 of the 10 languages, from 4% for German to 8% for Polish.

We perform a quantitative analysis by calculating the similarity between machine and human translated sentences in the validation set to discover if there are any correlations between these scores and experimental results for SPL and SPL (MT). The discussion on this is included in the supplementary document.

5.6 SPL (+English): Train with English

Similar to SPL but we also include augmented English dataset in the training data. Unlike the Bootstrap experiment, the results are worse for most languages in comparison with SPL results except for 3 languages, as shown in Table 5. This is expected since in the target language dataset, parameters have native values and match program parameters. Thus adding more English examples only helps if the translation quality of the NMT is much lower for the target language.

The drop in the accuracy is due to two main reasons: First, our training dataset for each foreign language is generated by machine translation. The outputs are more natural looking than synthesized English data even though they might not have correct grammar structure. Mixing distributions in train set is not a good approach specially since we validate and test on natural utterances. Second, the parameter values used in the foreign language are localized using our approach and are no longer in English. Thus seeing English examples with English parameters causes nothing but more confusion for the neural model's copying mechanism as we observe some English parameters present in the logical forms predicted by the model for the first 8 languages in Table 5.

5.7 SPL (Few Shot): Train with a Few Human Translations

In our final experiment, we concatenate the validation set obtained by humans to the training set generated using SPL. Since the validation size is

	SPL				SPL (MT)				SPL (+English)				SPL (Few Shot)			
Language	**Dev**		**Test**		**Dev (MT)**		**Test**		**Dev**		**Test**		**Dev (combined)**		**Test**	
	em	sm	em	sm	em	sm	em	sm	em	sm	em	sm	em	sm	em	sm
Arabic	68.88	69.79	65.88	67.94	74.62	74.92	**70.00**	**73.24**	72.51	73.41	66.76	70.00	74.92	75.68	67.94	69.41
German	63.14	63.44	65.29	65.88	60.73	62.24	61.76	62.94	56.19	59.52	56.76	60.00	67.82	68.28	**67.65**	**67.65**
Spanish	75.53	77.64	76.47	77.06	78.55	79.15	77.06	78.24	73.11	74.62	69.41	71.47	78.85	80.36	**79.12**	**80.29**
Persian	52.87	55.29	54.71	57.06	60.42	61.93	48.53	51.47	45.62	49.55	46.18	50.00	67.67	68.58	**66.18**	**67.94**
Finnish	68.88	69.49	68.82	69.41	59.21	59.21	61.47	62.35	54.68	55.89	56.47	57.06	68.43	69.64	**72.06**	**72.35**
Italian	78.25	78.25	77.65	77.94	73.41	73.41	73.53	73.53	71.98	71.98	73.24	73.82	77.79	77.79	**78.24**	**78.24**
Japanese	72.21	72.21	72.06	72.35	76.74	76.74	73.82	74.12	80.06	80.36	75.00	75.59	79.91	79.91	**76.47**	**76.76**
Polish	64.05	65.26	65.29	66.47	61.93	62.54	57.94	59.71	55.89	56.5	54.41	55.88	70.54	71.15	**67.35**	**67.94**
Turkish	67.67	67.67	65.59	65.59	56.8	57.4	57.35	57.94	66.16	67.37	65.00	66.76	74.47	74.92	**73.24**	**73.53**
Chinese	67.07	71.00	63.82	70.00	58.91	59.52	56.47	58.82	65.86	70.09	62.35	67.35	64.80	66.16	**64.41**	**68.24**
Arabic	71.69	71.96	70.84	70.84	76.46	76.46	72.05	72.05	71.43	71.43	**74.22**	**74.22**	76.98	77.12	73.98	73.98
German	74.34	74.34	73.98	74.7	76.72	76.72	71.57	72.05	74.60	74.87	75.18	76.63	81.35	81.35	**77.11**	**77.59**
Spanish	78.57	78.84	77.83	77.83	78.04	78.31	**81.2**	**82.41**	77.25	77.78	77.35	78.07	80.82	80.95	80.48	80.72
Persian	64.81	70.37	67.71	70.84	69.31	69.84	64.58	68.43	65.61	68.25	67.71	69.4	76.46	76.59	**75.66**	**76.39**
Finnish	72.75	72.75	68.67	68.92	70.11	70.11	69.88	70.60	68.78	69.05	67.71	67.95	80.95	81.22	**80.72**	**81.2**
Italian	78.04	78.04	78.07	78.07	78.84	78.84	78.31	78.31	80.16	80.16	**82.41**	**82.41**	83.60	83.60	79.52	79.52
Japanese	69.58	70.37	70.60	71.33	75.93	76.72	67.47	68.19	65.87	66.67	66.51	67.23	77.78	78.57	**77.59**	**78.55**
Polish	73.81	74.34	74.22	74.22	72.75	72.75	74.46	74.46	67.46	67.99	71.33	71.33	80.82	80.82	**79.28**	**79.28**
Turkish	73.28	73.28	78.07	78.07	75.4	75.93	80.00	80.00	80.69	80.69	81.20	81.20	81.35	81.35	**81.93**	**81.93**
Chinese	69.05	70.63	66.27	67.47	74.07	74.60	61.93	62.65	60.58	64.02	62.41	64.34	78.97	79.10	**76.63**	**76.63**

Table 5: Experiment results for hotels (top rows) and restaurants (bottom rows) domain using SPL. em and sm indicate exact and structure match accuracy respectively. All Dev and Test sets are human translated data except for Dev (MT) which is machine translated and Dev (mixed) where we combine machine and human translated data into one set. Exact match accuracies for English Test set are 69.00% for hotels, and 74% for restaurants.

much smaller than the training size (0.03% for hotels and 0.12% for restaurants) this is similar to a few-shot scenario where a small set of examples from evaluation and test distribution are used for training. We also create a combined validation set by concatenating the machine and human translated examples from validation set. We then train the semantic parser on the joint train set and evaluate on the mixed development set. We experimented by evaluating only on machine translated validation data too but the results did not improve.

As shown in Table 5, the test results are improved significantly across 9 languages for hotels domain achieving new state-of-the-art results on this benchmark. The improvement is partly due to low overlap between train and test distributions as is evident by the low BLEU scores (Papineni et al., 2002) in Table 4 and partly due to few-shot and transfer capabilities of transformer models (Brown et al., 2020; Campagna et al., 2020; Moradshahi et al., 2019) attained via heavy pretraining on large corpora from different domains.

This shows that a small, cheap addition of training data improves the model significantly. The exact match accuracy varies across languages, with a low of 64% for Chinese and a high of 79% for Spanish for hotels, and a low of 76% for Persian and a high of 82% for Turkish for restaurants. This compares favorably with the respective 69% and 74% accuracy obtained by the semantic parser trained on the original English dataset for English questions.

We have performed an error analysis on the results generated by the parser which is included in the supplementary material due to space constraints.

6 Conclusion

This paper presents a toolkit and methodology to extend and localize semantic parsers to any language at a fraction of the cost and human effort compared to previous methods. The toolkit can be used by any developer to extend the current capabilities of their QA system to a new domain and language in less than 24 hours leveraging mature public NMT systems. We found our approach to be effective on a recently proposed QA semantic parsing dataset which is significantly more challenging than other available multilingual datasets in terms of sentence complexity and ontology size.

Unlike prior work, our generated datasets are automatically annotated using English logical forms and require no human annotations. To minimize cost, we do parameter augmentation with local values after translation to increase coverage. Our model outperforms the baseline by improving human test accuracy ranging from 30% to 40% depending on the language.

Aside from creating new dataset and resources which will be released open-source upon publication, our methodology enables further investigation and creation of new benchmarks helpful to trigger more research on this topic.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, pages 1533–1544.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica S Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. *arXiv preprint arXiv:2005.00891*.

Giovanni Campagna, Silei Xu, Mehrad Moradshahi, Richard Socher, and Monica S. Lam. 2019. Genie: A generator of natural language semantic parsers for virtual assistant commands. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2019, pages 394–410, New York, NY, USA. ACM.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *arXiv preprint arXiv:2003.05002*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Deborah A Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the atis task: The atis-3 corpus. In *Proceedings of the workshop on Human Language Technology*, pages 43–48. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.

Long Duong, Hadi Afshar, Dominique Estival, Glen Pink, Philip Cohen, and Mark Johnson. 2017. Multilingual semantic parsing and code-switching. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 379–389, Vancouver, Canada. Association for Computational Linguistics.

Christian Federmann, Oussama Elachqar, and Chris Quirk. 2019. Multilingual whispers: Generating paraphrases with translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 17–26, Hong Kong, China. Association for Computational Linguistics.

Carolin Haas and Stefan Riezler. 2016. A corpus and semantic parser for multilingual natural language querying of OpenStreetMap. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 740–750, San Diego, California. Association for Computational Linguistics.

Jonathan Herzig and Jonathan Berant. 2018. Decoupling structure and lexicon for zero-shot semantic parsing. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

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 generalization. *arXiv preprint arXiv:2003.11080*.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany. Association for Computational Linguistics.

Rohit J Kate, Yuk Wah Wong, and Raymond J Mooney. 2005. Learning to transform natural to formal languages. In *AAAI*, pages 1062–1068.

Mehrad Moradshahi, Hamid Palangi, Monica S. Lam, Paul Smolensky, and Jianfeng Gao. 2019. Hubert untangles bert to improve transfer across nlp tasks.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 1073–1083.

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.

Tom Sherborne, Yumo Xu, and Mirella Lapata. 2020. Bootstrapping a crosslingual semantic parser.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a semantic parser overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1332–1342. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

William A. Woods. 1977. Lunar rocks in natural english: explorations in natural language question answering.

Silei Xu, Giovanni Campagna, Jian Li, and Monica S Lam. 2020. Schema2QA: Answering complex queries on the structured web with a neural model. *arXiv preprint arXiv:2001.05609*.

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*, pages 3911–3921.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, page 1050–1055. AAAI Press.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.