Align and Augment:
Generative Data Augmentation for Compositional Generalization

Francesco Cazzaro
Universitat Politècnica
de Catalunya
Barcelona, Spain
francesco.cazzaro@upc.edu

Davide Locatelli
TORTUS AI
London, UK
davide@tortus.ai

Ariadna Quattoni
Universitat Politècnica
de Catalunya
Barcelona, Spain
aquattoni@cs.upc.edu

Abstract

Recent work on semantic parsing has shown that seq2seq models find compositional generalization challenging. Several strategies have been proposed to mitigate this challenge. One such strategy is to improve compositional generalization via data augmentation techniques. In this paper we follow this line of work and propose ARCHER, a data-augmentation strategy that exploits alignment annotations between sentences and their corresponding meaning representations. More precisely, we use alignments to train a two step generative model that combines monotonic lexical generation with reordering. Our experiments show that ARCHER leads to significant improvements in compositional generalization performance.

1 Introduction

Semantic parsing is the task of mapping natural language sentences (NLs) to their corresponding meaning representations (MRs). Sequence-to-sequence (seq2seq) transformers based on encoder-decoder architectures have become predominant for this task and have shown impressive performance (Banerjee et al., 2022; Yin et al., 2021; Kamath and Das, 2019). However, seq2seq models have been shown to have a limited compositional generalization ability (Keysers et al., 2020; Lake and Baroni, 2018).

One natural approach to improve compositional generalization is to feed seq2seq models with additional data, increasing the set of observed patterns (Qiu et al., 2022a; Akyürek et al., 2021; Andreas, 2020). The additional data is assumed to be automatically generated from the available training set using a generation strategy: this is usually referred to as *data augmentation*.

In this paper we follow this line of research and propose ARCHER: Align and Augment foR Compositional Hard GEneRalization. ARCHER is a data augmentation strategy that utilizes word alignments between NL and MR pairs. In a first step, a recursive model generates monotonically aligned NL/MR pairs. In a second step, a reordering model rearranges symbols in the MRs, ensuring correct alignment with the NLs. This combines the strengths of traditional recursive models, which excel at modelling sequence distributions, and seq2seq architectures, which excel at inducing arbitrary features of the input and output sequences.

We evaluate ARCHER on two multilingual datasets annotated with word alignments: GEOALIGNED (Locatelli and Quattoni, 2022), an extension of the GEO benchmark, and ATISALIGNED, which we introduce as part of our research, similarly extending the ATIS benchmark. Our experiments demonstrate that ARCHER significantly enhances the compositional generalization capabilities of seq2seq semantic parsers. In the English GEO dataset's *length* partition, with ARCHER data a parser accuracy almost doubles to 46%. Similarly, in the *query* partition, performance improve from 72% to 82%.

Compared to alternative augmentation approaches, ARCHER leads to higher improvements in compositional generalization, especially on the most challenging *length* partitions.

The contributions of our work are:

- We introduce ARCHER, a new data augmentation technique that utilizes word alignments with a two-step generative process.
- Our approach significantly improves compositional generalization in seq2seq models, with remarkable improvements on *length* splits.
- An analysis of the data generated by ARCHER shows that it can produce more accurate and diverse samples than alternative approaches.
- As a side contribution, we introduce ATISALIGNED, a semantic parsing dataset aug-

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics Volume 1: Long Papers, pages 369–383
March 17-22, 2024 ©2024 Association for Computational Linguistics

mented with word alignment annotations[1].

2 Related Work

Data Augmentation. Various works have explored data augmentation within the context of semantic parsing. Some methods have recombined samples by softly interpolating input/output examples (Guo et al., 2020), utilizing rules to swap tokens appearing in similar contexts (Andreas, 2020) or by transformations based on symmetries (Akyurek and Andreas, 2023). Other approaches used grammars for sampling, such as SCFG (Jia and Liang, 2016; Oren et al., 2021) or QCFG (Qiu et al., 2022a). Recombined data has also been obtained through subtree substitutions (Yang et al., 2022; Li et al., 2023), prototype-based generative models for recombination and resampling (Akyürek et al., 2021), or through the exploitation of crosslingual datasets (Rosenbaum et al., 2022). Other approaches have focused on generating an MR first, followed by the use of a generative model to predict an associated utterance (Zhong et al., 2020; Tran and Tan, 2020; Wang et al., 2021b). However, different from the focus of our work, these last three approaches were not tested on compositional generalization.

Compositional Generalization. Recently, researchers have raised the question of whether models can perform compositional generalization (Lake and Baroni, 2018; Finegan-Dollak et al., 2018; Kim and Linzen, 2020). The general consensus within the community is that sequence to sequence models struggle significantly in this aspect (Loula et al., 2018; Keysers et al., 2020; Kim and Linzen, 2020). One approach to test compositional generalization is to train a semantic parser on sequences up to a fixed length and test it on longer ones, forcing the model to predict novel combinations (commonly referred as the *length* partition). This is a challenging task, similar to how traditional grammatical inference algorithms are tested in the formal language community. The fact that seq2seq models fail at this type of generalization has been widely documented (Anil et al., 2022). Further studies have suggested that employing large pretrained language models does not appear to aid compositional generalization (Oren et al., 2020; Qiu et al., 2022b), and that both structural (Bogin et al., 2022) and length factors make it particularly challenging. While compositional generalization has mostly been studied in the context of semantic parsing, it has also been observed that models struggle with it in other tasks (Yao and Koller, 2022). Consequently, these findings have spawned a plethora of works dedicated to improving compositional generalization performance (Li et al., 2019; Liu et al., 2020a; Gordon et al., 2020; Chen et al., 2020; Nye et al., 2020; Oren et al., 2020; Zheng and Lapata, 2021; Conklin et al., 2021; Shaw et al., 2021; Csordás et al., 2021; Liu et al., 2021a; Zheng and Lapata, 2022; Weißenhorn et al., 2022; Jambor and Bahdanau, 2022; Lindemann et al., 2023b; Zheng et al., 2023; Yin et al., 2023). In this context, it has been observed that alignments are highly valuable for compositional generalization (Shi et al., 2020), and it has been suggested that parsers may be hindered by the lack of alignment usage (Zhang et al., 2019). As a result, efforts have been made to create datasets with alignment annotations (Shi et al., 2020; Herzig and Berant, 2021; Locatelli and Quattoni, 2022) and numerous models have been proposed to leverage alignment information (Lei et al., 2020; Wang et al., 2021a; Herzig and Berant, 2021; Liu et al., 2021b; Sun et al., 2022; Cazzaro et al., 2023; Lindemann et al., 2023a).

3 Preliminaries

This section introduces the preliminary background on word alignments and Weighted Finite state Automata (WFA) necessary to understand ARCHER.

3.1 Word alignments

We assume that we are given a pair of sequences $(\mathbf{x}, \mathbf{y})$ where $\mathbf{x} = x_1, \dots, x_n$ is a sequence of n tokens and $\mathbf{y} = y_1, \dots y_m$ is a sequence of m tokens. Because the concept of token alignments was originally developed in the context of machine translation, tokens are usually referred as words.

Formally, a word alignment $\mathcal{A}$, is defined as a set of bi-symbols, where each bi-symbol $b = (x_i, y_j)$ pairs the i-th word in x with the j-th symbol in y. If a word x_i is not aligned to any word in y, then it is aligned to a special symbol ε and the resulting bi-symbol is denoted by (x_i, ε). Similarly, if a word y_j is not aligned to any word in $\mathbf{x}$, this will be denoted with the bi-symbol (ε, y_j) [2]

[1] The dataset is available at https://github.com/interact-erc/AtisAligned.git

[2] Note that this framework allow for 1-to-many and many-to-1 alignments. For example, if we wish to align words x_i, x_j to a single word y_k we can choose a 'head word' among the x pair and align the 'non-head' words to ε. In practice, annotators have shown a large degree of agreement in their

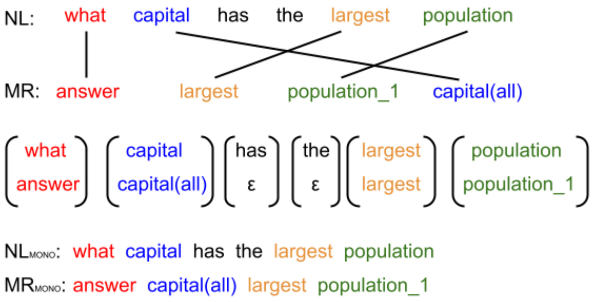
NL: what capital has the largest population
MR: answer largest population_1 capital(all)
what
answer
capital
capital(all)
has
ε
the
ε
largest
largest
population
population_1
NLMONO: what capital has the largest population
MRMONO: answer capital(all) largest population_1

Figure 1: Example of a sample from the GEOALIGNED dataset. From top to bottom: first is shown the NL/MR pair with the corresponding alignments, then the associated bi-symbols sequence and finally the NL/MR pair reordered monotonically. Notice that the NL remains identical while the MR is in a different order.

In the case of semantic parsing the sequence pair $(\mathrm{NL}, \mathrm{MR})$ will consist of a natural language sentence and its corresponding meaning representation. Hence, NL denotes a sequence of words and MR a sequence of meaning representation symbols.

A pair of aligned sequences can be mapped to a sequence of bi-symbols, this is achieved by fixing the order of one of the two sequences and re-ordering the second sequence according to the alignments.

For example, take the pair of sequences $x = ABCD$ and $y = FGH$ and suppose word A is aligned to F, word B to H, word C to G and word D is not aligned. Keeping the order of x fixed, this alignment will be mapped to the sequence of bi-symbols $[(A, F), (B, H), (C, G)(D, \varepsilon)]$. If we extract the x words from the bi-symbols sequence we obtain $x = ABCD$ but extracting the words of y would result in $y = FHG$, where the words H and G have been re-ordered (Figure 1).

For our semantic parsing data-augmentation strategy we will be learning a generative model of aligned NL/MR bi-symbol sequences. In this case we will maintain the order of the NL sentence but the order of the MR symbols might differ from their original MR order.

In fact, it is easy to see that for all NL/MR pairs that are not monotonically aligned the mapping to a sequence of bi-symbols will result in at least one reordering of the MR sequence.

3.2 WFA

A Weighted Finite Automata over an alphabet Σ is defined as a tuple $A = \{\alpha_1, \alpha_\infty, \{A_\sigma\}\}$ where $A_\sigma \in \mathbb{R}^{n\times n}$ is the transition matrix associated to each symbol $\sigma \in \Sigma$ and α_1, $\alpha_\infty \in \mathbb{R}^n$ are the initial and final weight vectors. Given a sequence $x = x_1, \ldots, x_n$ where $x_i \in \Sigma$ a WFA realizes the function:

$$f_A(x) = \alpha_1^\top A_{x_1}, \cdots, A_{x_n} \alpha_\infty \tag{1}$$

A WFA is a recurrent neural network with linear activation function, this equivalence has been proven in Rabusseau et al. (2019).

Due to the linearity of the activation function, the parameters of this subclass of RNNs can be estimated in closed-form via what is usually referred as the spectral method. For more details on WFAs and their training algorithms we refer the reader to Balle et al. (2014). We also implement the optimizations described in Quattoni et al. (2017).

Probabilistic finite state automata are a subclass of WFAs, thus WFAs can be used to model sequence distributions. In this case, the learning algorithm is designed to minimize an l_2 loss function over the observed sub-sequence expectations. That is why when spectral learning is used to estimate a (probabilistic) sequence distribution it is usually described as moment-matching. This nomenclature refers to the fact that the loss function will attempt to match the empirical sub-sequence distribution observed in training. For a more detailed description of WFAs in the context of language modeling, as well as comparisons to other models, we refer the reader to Quattoni and Carreras (2019).

We conducted preliminary experiments in which we explored the possibility of modeling the bi-symbol sequence distribution with Transformers and LSTMs. However, we found it challenging due to calibration problems (Desai and Durrett, 2020). We also experimented with simpler ngram models, which not surprisingly also failed since these models are unable to make proper generalizations from relatively small training sets. As a result, we made the decision to use WFAs to model the bi-symbol sequence distribution. This seemed like the natural choice since moment matching is specifically designed for density estimation.

We suspect that the difficulty in performing density estimation with other deep sequence model architectures might explain why generative data augmentation via explicit use of word alignments has not been attempted before in the literature. That being said, it is important to note that the cornerstone of our data-augmentation strategy is to model and sample from the (aligned) bi-symbol sequence

choices of head words (Locatelli and Quattoni, 2022)

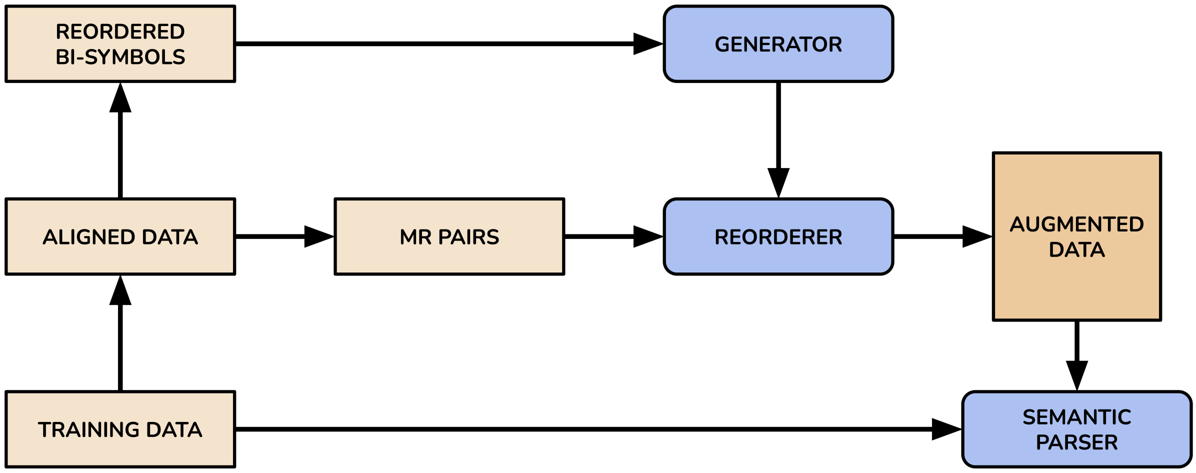

Figure 2: Schematic illustration of the ARCHER data augmentation approach. We begin with a set of *training data* and utilize word alignment information to extract *aligned data*. This aligned data is used to train our *Generator* and *Reorderer* models. The Generator model is trained on NL/MR sequences of *reordered bi-symbols*. On the other hand, the Reorderer model is trained on *MR pairs*, which consist of the original MRs and their corresponding monotonically aligned versions. The Generator outputs data that is then passed through the Reorderer, resulting in *augmented data*. This augmented data, along with the training data, is fed into the *semantic parser*.

distribution. Consequently, we believe it is worth exploring other density estimation methods to learn sequence distributions. However, this is outside the scope of this paper and we leave it for future work.

4 Data augmentation with ARCHER

In this section we present our main contribution, ARCHER: a two-step data augmentation approach that improves compositional generalization by leveraging a generative recursive sequence model over aligned bi-symbols. Figure 2 provides a graphical illustration of our approach.

We assume that we are given a training set T consisting of NL/MR pairs (x, y) which have been annotated with word-alignments, mapping NL words to MR symbols (described in the previous section). Our objective is to create an additional training set T' by generating new samples (x', y'). We will then train a semantic parser using the original training set T augmented with the additional samples in T'. Ideally, the generation process should create novel patterns that will improve the compositional generalization of the default semantic parser.

Generator. We start by reordering aligned NL/MR pairs to enforce a monotonic alignment between the NLs and MRs tokens. By applying this transformation to all training pairs in T, we obtain a dataset $T_{monotonic}$ consisting of sequences of bi-symbols and we use it to train a generative model of the bi-symbol distribution. We can then sample from the learned distribution and generate new bi-symbol sequences.

We choose to model this distribution using WFAs for two main reasons: (1) WFAs are defined as generative models and can naturally model the prefix distribution necessary for generation; (2) since they are recursive they seem the natural choice to generate longer sequences from a distribution estimated from short sequences. This is important because our focus is on compositional generalization, which requires the ability to recombine known elements to create longer novel structures. This being said, ARCHER is a general approach and the essence of the idea is to model and sample from the bi-symbol distribution estimated from aligned data: in this sense other models could also be used to model the bi-symbol distribution.

We now turn our attention to some details on how we train and sample from the WFA. When training the WFA we append special beginning <BOS> and end of sequence symbols <EOS> to every sequence of bi-symbols. To generate a sequence we initialize the process with the <BOS> prefix. We continue to generate new bi-symbols b_i by sampling from the conditional distribution $P_{\mathrm{WFA}(b_i|b_{1:i-1})}$, where $b_{1:i}$ refers to the prefix: $[b_1, b_2, \ldots, b_i]$. In practice, when sampling we only consider the top-k bi-symbols with highest probability.

In principle, the generation stops when the special <EOS> symbol is generated. However, in order to bias the process to produce longer sequences we fix the conditional probability of <EOS> to 0 until a desired minimum length t is reached. After generating t bi-symbols we reset the <EOS> probability to its true value and continue sampling. In other words, the sample is never cut abruptly. In appendix C we look at the generation without the t constraint including experimental results.

After we generate an initial set of samples we remove all duplicates and samples present in T. Finally, we observed that simple filtering strategies can further improve the quality of the generated samples, this is described in more detail in 4.1.

Reorderer. From the generated bi-symbols we can extract an NL/MR pair by simply removing all epsilons. However, since the bi-symbol distribution was trained over a transformed dataset, i.e. $T_{monotonic}$, the symbols in the MR might not be in the correct order and cannot be directly used for data augmentation. To address this problem we use the word-aligned data to train a re-orderer model which takes an unordered MR sequence and outputs it in the correct order. The re-orderer model is trained from pairs $(\mathrm{MR}, \mathrm{MR}_{mono})$, where MR is the original sequence and MR_{mono} is the sequence obtained after the transformation that enforces a monotonic alignment between the MR and its corresponding NL.

More specifically, for the re-orderer model we train a standard encoder-decoder mBART model. At decoding time we do not impose any constraints in the output generation. That is, we don't enforce that the output sequence has to be a permutation of the input. We don't even require that the re-ordered MR has the same length as the input MR. Thus the reordering model is free to add, substitute and delete symbols of the input MR.

In preliminary experiments we observed that when given this freedom, the re-orderer model could rectify some errors made by the generative bi-symbol model, errors in structure that went beyond symbol re-ordering. We also experimented with a constrained decoding strategy that restricts the outputs to be permutations of the input, however no significant gain was observed (appendix B).

After running the re-orderer model over the sequences sampled from the learned bi-symbol distribution we obtain the final sequences T' that augment the original sequences in T. Both sets are then used to train the final semantic parser.

4.1 Filters

As expected, the data generation process is not error-free and will generate some malformed NL/MR pairs. Errors can be of different types: e.g. the NL might be malformed, the MR might be malformed or they might be both independently correct but the combination might be wrong. To improve the quality of the generated samples we experimented with different filtering strategies.

Given that our bi-symbol generator is a density estimator, we can compute the probability assigned to each generated sample. We can then filter out samples whose probability is lower than a certain threshold. Alternatively, we could also train additional density estimators for the NL and the (monotonically transformed) MR sequences separately. We could then score a sample based on the probability given by the independent NL or MR models.

It is important to note that although filtering generated samples based on their probability might seem natural, it has an important limitation. If we where to select only the most probable samples, we run the obvious risk of generating an augmented set of low sample diversity (relative to the original training set) that will add no useful novel information. Therefore, there is always a trade-off between the correctness and diversity of the augmented data.

To complement the previous distributional strategies, we also considered a different approach based on using the re-orderer model for detecting badly formatted MRs. Recall that the re-orderer is unconstrained and can add, delete or substitute symbols of the generated MR. We observed that while a few corrections might fix some errors of the generator, a large deviation in the number of symbols between the original and the reordered MR tends to signal that the generated MR is badly formed. Therefore with an appropriate threshold this can be used to filter out badly generated samples.

In the experiments we validate the choice of filter on a development set. Overall, the re-ordering filter was the best for most data-sets and partitions.

5 Experimental setup

5.1 Datasets

We evaluate our data augmentation approach on two widely-used semantic parsing benchmarks: the multilingual GEO dataset and the English ATIS. Both of these datasets define two standard benchmarks that are used to evaluate compositional generalization: (1) the *query* partition, introduced by

Finegan-Dollak et al. (2018), is designed to be compositional by ensuring that the templates of the test set MRs are never seen during training; (2) the *length* partition, introduced by Herzig and Berant (2021), assigns the longest MR sequences to the test. The *length* partition is known to be the most challenging and it could be argued that is the most rigorous, since it forces the parser to learn proper recursions. In fact, this type of evaluation mimics the classical way in which language models are evaluated in the formal language community. The statistics of the datasets after augmentation are detailed in appendix D.

GEOALIGNED. Locatelli and Quattoni (2022) extended the popular GEO dataset (Zelle and Mooney, 1996) with word alignment annotations. The dataset contains 880 questions about US geography annotated with MRs in the FunQL formalism (Kate et al., 2005). It is available in three languages: English, Italian, and German, providing a multilingual aspect to our evaluation. We follow Wang et al. (2021a) in removing brackets.

ATISALIGNED. The original ATIS dataset Dahl et al. (1994) revolves around flight booking queries in English and contains 5409 samples. We use the FunQL formalism. We have augmented the dataset with word alignment annotations and made it publicly available. We also removed brackets from the MRs. Appendix A includes more details.

5.2 Semantic parsing model

As a base semantic parser we use a sequence-to-sequence transformer model: MBART (Liu et al., 2020b). This is the multilingual version of BART and has been shown to give state-of-the-art performance for semantic parsing (Bevilacqua et al., 2021). We validate hyper-parameters on the development set and all the results reported are the average of multiple runs.

5.3 Data augmentation techniques

ARCHER. Our data augmentation technique presented in section 4. We normally refer to ARCHER as using the ground truth alignments, however we also experiment with automatically induced alignments obtained with IBM model 5 (Brown et al., 1993). We refer to the setting with auotomatic alignments as ARCHER$_{\text{IBM}}$. Both the hyper-parameters of the WFA and the MBART re-orderer were validated on the dev set.

SELF-TRAINING. As a strong baseline, we consider a self-training approach. One of the motivations of this baseline is to evaluate what can be gained from self-training alone (i.e. without leveraging word-aligments). In this approach we use the original training data to train four models:

1. A decoder-transformer trained on NL sequences.
2. A decoder trained on MR sequences.
3. A seq2seq encoder-decoder that takes NL sequences and predicts their corresponding MRs. This is essentially the base semantic parser trained on the original data only.
4. A seq2seq encoder-decoder that takes an MR and predicts a corresponding NL. This is also trained using the original data but swapping inputs and outputs.

With these four models we can test two self-training strategies: generating an NL using the NL encoder first and predicting its corresponding MR using the NL to MR encoder-decoder; and generating an MR using the MR encoder first and then predicting its corresponding NL.

For each dataset and partition, we validated the best self training strategy on the development set. We also applied and validated the filtering strategies. The self-training results reported in the next section correspond to the best sampling strategy and filter (chosen in development). In preliminary experiments we also tried WFAs for models 1) and 2) but without any significant improvements.

GECA Andreas (2020). A method for data augmentation based on identifying fragments of training examples that appear in similar contexts and recombining them to generate new data.

SCFG Jia and Liang (2016). A method for obtaining data recombination using an induced synchronous context-free grammar.

SUBS Yang et al. (2022). Based on subtree substitution for compositional data augmentation.

5.4 Evaluation

For evaluation we use the standard exact match accuracy: the prediction is correct if the predicted MR is the same as gold.

Model	GEO						ATIS		AVG
	EN		IT		DE		EN		-
	Q	LEN	Q	LEN	Q	LEN	Q	LEN	-
mBART	72.36	27.50	76.59	23.33	56.30	18.20	62.15	28.71	45.64
+ GECA	**87.64**	29.16	**83.57**	30.83	65.12	22.97	61.10	27.69	51.01
+ Self-Training	77.07	27.37	81.46	28.21	64.38	23.93	**64.91**	26.25	49.20
+ SCFG	73.41	31.07	74.47	28.09	59.02	20.23	-	-	-
+ SUBS	79.74	43.03	78.78	28.80	65.36	25.59	-	-	-
+ ARCHER	82.11	**46.31**	82.43	**38.33**	**72.68**	**31.19**	63.31	**29.79**	**55.77**
+ ARCHER$_{\text{IBM}}$	81.30	44.16	79.02	30.59	69.51	29.16	62.92	29.63	53.27

Table 1: Exact-match accuracy scores on all compositional partitions. Q stands for the query partition and LEN for the length partition. The last column, AVG, reports the average of all scores as a single aggregation metric.

6 Results

Table 1 shows the performance of the different data augmentation techniques on all datasets and compositional partitions. We start by examining the performance in the length partition (LEN). ARCHER outperforms the other methods significantly and exhibits a substantial improvement over the base semantic parser. In contrast, the other methods obtain rather moderate improvements with the exception of SUBS in English.

Looking at the query partition (Q), we observe that all the data augmentation techniques lead to significant improvements over the base semantic parser with the only exception being SCFG. For this partition there doesn't seem to be a clear winner and the different techniques seem to perform similarly. The only exceptions being GEO-EN for which GECA is significantly better and GEO-DE for which Archer is significantly better.

From this experiment we conclude that ARCHER is an effective data augmentation technique that can significantly improve the compositional generalization of seq2seq models, especially in length generalization. These results show that a recursive generative model can successfully leverage aligned data and generate samples that are both diverse and accurate. Section 7 further complements these conclusions by evaluating directly the correctness and diversity of the different augmentation strategies.

Finally, in Table 2, we present the results for the standard IID partitions. These partitions are less challenging and do not require compositional generalization. As expected, the data-augmentation techniques designed to improve compositional generalization do not have any significant impact. The simple IID partition does not benefit by seeing novel recombinations since most templates in the test partition are observed in the original training partition. Note that the generation is not perfect and some generated samples can contain errors that our filtering methods fail to detect. Therefore we hypothesize that in this case the errors that we introduce are not counterbalanced by the benefits of our approach and thus we might have some minor drop in performance.

Model	GEO			ATIS
	EN	IT	DE	EN
mBART	87.38	86.67	75.50	85.26
+ GECA	87.49	87.50	74.76	83.02
+ Self-Training	88.33	85.47	75.23	84.96
+ SCFG	84.40	83.69	73.45	-
+ SUBS	85.83	84.28	73.39	-
+ ARCHER	86.42	82.50	74.52	84.37
+ ARCHER$_{\text{IBM}}$	86.60	82.47	74.46	84.15

Table 2: Exact-match accuracy on the IID partitions of GEOALIGNED and ATISALIGNED.

7 Analysis of Generated Data

In this section, we analyze the quality of the data generated by three different strategies on GEOALIGNED. We focus on evaluating the correctness and diversity of the generated samples. Correctness ensures accurate representation of desired patterns, enhancing reliability. Diversity aids compositional generalization, allowing the model to handle novel combinations effectively. Refer to Table 8 in Appendix E for some examples of

ARCHER generations.

7.1 Methodology

Sampling. To analyze data quality we randomly select samples of different lengths (ranging from 7 to 11). More precisely we select 20 samples for each length. In total we will evaluate 100 generated samples for each data augmentation technique.

Correctness. Two annotators[3] rated the quality of the generated NL/MR pairs by answering the following questions:

1. Is the natural language sentence correct? (**NL** column in Table 3)
2. Is the meaning representation correct? (**MR** column)
3. Is the combined NL/MR pair correct? (**BOTH** column)

We instructed annotators to label nonsensical sentences such as "How many people live in a river?" as correct, since its semantic incorrectness can only be deduced from world knowledge. Since this might be seen as a soft definition of correctness, annotators were also asked: Is the pair semantically correct? (**SEM** column). Although this annotation task might seem complex, annotators showed a high degree of agreement, disagreeing on around 15% of the examples. Each disagreement was discussed and resolved by reaching a consensus.

Diversity. For all samples in which both the NL and the MR were correct, we measure diversity using the BLEU metric (Papineni et al., 2002). BLEU scores range from 0 to 1, indicating similarity between a target sequence and a reference set. We assess diversity in two ways: inter-diversity (comparing samples to the training data) and intra-diversity (examining diversity within the generated set).

To calculate both diversity measures, we compare each generated sample against all other samples in the reference set. The highest BLEU score is recorded, and the average score across all generated samples is calculated. By using the maximum BLEU score, we capture the closest similarity between the generated sample and the reference samples. The final diversity score is obtained by subtracting the average score from 1. We run this procedure on the NLs and the MRs separately so that we can estimate both the diversity of natural language sentences and meaning representations.

Approach	NL	MR	BOTH	SEM
GECA	0.49	0.48	0.4	0.36
Self-Training	**0.66**	0.26	0.26	0.25
ARCHER	0.45	**0.61**	**0.43**	**0.38**

Table 3: Proportion of GEO augmented examples labeled as correct by all annotators.

Approach	NL		MR	
	Inter	Intra	Inter	Intra
GECA	0.31	0.39	0.45	0.49
Self-Training	0.35	0.42	**0.54**	**0.61**
ARCHER	**0.43**	**0.52**	0.52	0.58

Table 4: Diversity scores of GEO augmented examples.

7.2 Overview of diversity and correctness

Table 3 presents the results of our data augmentation correctness analysis. ARCHER generates the most correct sample pairs (**BOTH**) and the best MRs. For SEM, GECA closely trails ARCHER, suggesting that both methods successfully capture contextual information related to the recombined elements. Notably, Self-Training outperforms other approaches in NL correctness, likely due to its utilization of pre-trained embeddings, which provides a natural advantage in generating coherent NLs.

Table 4 shows the diversity scores for the three augmentation methods. ARCHER demonstrates significantly higher inter- and intra-diversity. This is most evident in the NL scores. In terms of MRs, Self-Training produces more diverse samples, but ARCHER lags behind by just 0.03 points. Nevertheless, when considering correctness (Table 3), it is evident that a majority of the Self-training MRs are incorrect, thus showing that ARCHER offers the best trade-off of correctness and diversity. Overall, considering both correctness and diversity, our analysis shows that ARCHER yields better samples.

8 Conclusion

This paper introduced ARCHER, a novel data augmentation method that utilizes word alignment information in a two-step process. First, it generates

[3] As annotators we chose students that had previous experience with the datasets since the annotation task is not trivial. To render the process unbiased, we shuffled samples from the different generations methods, so that an annotator had no way of telling which method produced a specific sample.

a (monotonically aligned) NL/MR pair, then it reorders the MR. We evaluated our method on multilingual semantic parsing datasets and observed consistent improvements in the compositional generalization of the base semantic parser, especially in length generalization. We also presented a complementary analysis of the generated data that showed that ARCHER generates more accurate and diverse samples than other augmentation techniques.

Limitations

One limitation of ARCHER is that it is relatively computationally demanding to run, since it involves training multiple models on top of the base semantic parser, including the generator, the reorderer, as well as models for the filters. While this is true also for the self-training baseline that we compared with, GECA is a simpler rule-based approach that does not require as many computational resources.

Another limitation of our work is that we focused solely on the FunQL formalism for the MRs. Future research should explore the application of the ARCHER technique to additional datasets, in order to determine if the performance improvements observed are consistently applicable across different formalisms. The reason why we primarily focused on FunQL is partly due to the scarcity of word alignment annotations available for semantic parsing datasets in alternative formalisms.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 853459. The authors gratefully acknowledge the computer resources at ARTEMISA, funded by the European Union ERDF and Comunitat Valenciana as well as the technical support provided by the Instituto de Física Corpuscular, IFIC (CSIC-UV). This research is supported by a recognition 2021SGR-Cat (01266 LQMC) from AGAUR (Generalitat de Catalunya).

References

Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. 2021. Learning to recombine and resample data for compositional generalization. In *International Conference on Learning Representations*.

Ekin Akyurek and Jacob Andreas. 2023. LexSym: Compositionality as lexical symmetry. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 639–657, Toronto, Canada. Association for Computational Linguistics.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. Exploring length generalization in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 38546–38556. Curran Associates, Inc.

Borja Balle, Xavier Carreras, Franco M. Luque, and Ariadna Quattoni. 2014. Spectral learning of weighted automata. *Machine Learning*, 96(1):33–63.

Debayan Banerjee, Pranav Ajit Nair, Jivat Neet Kaur, Ricardo Usbeck, and Chris Biemann. 2022. Modern baselines for sparql semantic parsing. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2260–2265, New York, NY, USA. Association for Computing Machinery.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One SPRING to Rule Them Both: Symmetric AMR Semantic Parsing and Generation without a Complex Pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

Ben Bogin, Shivanshu Gupta, and Jonathan Berant. 2022. Unobserved local structures make compositional generalization hard. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2731–2747, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, Robert L Mercer, et al. 1993. The mathematics of statistical machine translation: Parameter estimation.

Francesco Cazzaro, Davide Locatelli, Ariadna Quattoni, and Xavier Carreras. 2023. Translate first reorder later: Leveraging monotonicity in semantic parsing. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 227–238, Dubrovnik, Croatia. Association for Computational Linguistics.

Xinyun Chen, Chen Liang, Adams Wei Yu, Dawn Song, and Denny Zhou. 2020. Compositional generalization via neural-symbolic stack machines. In *Advances in Neural Information Processing Systems*, volume 33, pages 1690–1701. Curran Associates, Inc.

Henry Conklin, Bailin Wang, Kenny Smith, and Ivan Titov. 2021. Meta-learning to compositionally generalize. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3322–3335, Online. Association for Computational Linguistics.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. 2021. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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 *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-SQL evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.

Jonathan Gordon, David Lopez-Paz, Marco Baroni, and Diane Bouchacourt. 2020. Permutation equivariant models for compositional generalization in language. In *International Conference on Learning Representations*.

Demi Guo, Yoon Kim, and Alexander Rush. 2020. Sequence-level mixed sample data augmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5547–5552, Online. Association for Computational Linguistics.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Jonathan Herzig and Jonathan Berant. 2021. Span-based semantic parsing for compositional generalization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 908–921, Online. Association for Computational Linguistics.

Dora Jambor and Dzmitry Bahdanau. 2022. LAGr: Label aligned graphs for better systematic generalization in semantic parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3295–3308, Dublin, Ireland. Association for Computational Linguistics.

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.

Aishwarya Kamath and Rajarshi Das. 2019. A survey on semantic parsing. In *Automated Knowledge Base Construction (AKBC)*.

Rohit J. Kate, Yuk Wah Wong, and Raymond J. Mooney. 2005. Learning to transform natural to formal languages. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*, AAAI'05, page 1062–1068. AAAI Press.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882. PMLR.

Wenqiang Lei, Weixin Wang, Zhixin Ma, Tian Gan, Wei Lu, Min-Yen Kan, and Tat-Seng Chua. 2020. Re-examining the role of schema linking in text-to-SQL. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6943–6954, Online. Association for Computational Linguistics.

Yuanpeng Li, Liang Zhao, Jianyu Wang, and Joel Hestness. 2019. Compositional generalization for primitive substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4293–4302, Hong Kong, China. Association for Computational Linguistics.

Zhaoyi Li, Ying Wei, and Defu Lian. 2023. Learning to substitute spans towards improving compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2791–2811, Toronto, Canada. Association for Computational Linguistics.

Matthias Lindemann, Alexander Koller, and Ivan Titov. 2023a. Compositional generalisation with structured reordering and fertility layers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2172–2186, Dubrovnik, Croatia. Association for Computational Linguistics.

Matthias Lindemann, Alexander Koller, and Ivan Titov. 2023b. Compositional generalization without trees using multiset tagging and latent permutations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14488–14506, Toronto, Canada. Association for Computational Linguistics.

Chenyao Liu, Shengnan An, Zeqi Lin, Qian Liu, Bei Chen, Jian-Guang Lou, Lijie Wen, Nanning Zheng, and Dongmei Zhang. 2021a. Learning algebraic recombination for compositional generalization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1129–1144, Online. Association for Computational Linguistics.

Qian Liu, Shengnan An, Jian-Guang Lou, Bei Chen, Zeqi Lin, Yan Gao, Bin Zhou, Nanning Zheng, and Dongmei Zhang. 2020a. Compositional generalization by learning analytical expressions. In *Advances in Neural Information Processing Systems*, volume 33, pages 11416–11427. Curran Associates, Inc.

Qian Liu, Dejian Yang, Jiahui Zhang, Jiaqi Guo, Bin Zhou, and Jian-Guang Lou. 2021b. Awakening latent grounding from pretrained language models for semantic parsing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1174–1189, Online. Association for Computational Linguistics.

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

Davide Locatelli and Ariadna Quattoni. 2022. Measuring alignment bias in neural seq2seq semantic parsers. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 200–207, Seattle, Washington. Association for Computational Linguistics.

João Loula, Marco Baroni, and Brenden Lake. 2018. Rearranging the familiar: Testing compositional generalization in recurrent networks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 108–114, Brussels, Belgium. Association for Computational Linguistics.

Maxwell Nye, Armando Solar-Lezama, Josh Tenenbaum, and Brenden M Lake. 2020. Learning compositional rules via neural program synthesis. In *Advances in Neural Information Processing Systems*, volume 33, pages 10832–10842. Curran Associates, Inc.

Inbar Oren, Jonathan Herzig, and Jonathan Berant. 2021. Finding needles in a haystack: Sampling structurally-diverse training sets from synthetic data for compositional generalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10793–10809, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Inbar Oren, Jonathan Herzig, Nitish Gupta, Matt Gardner, and Jonathan Berant. 2020. Improving compositional generalization in semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2482–2495, Online. Association for Computational Linguistics.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Linlu Qiu, Peter Shaw, Panupong Pasupat, Pawel Nowak, Tal Linzen, Fei Sha, and Kristina Toutanova. 2022a. Improving compositional generalization with latent structure and data augmentation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4341–4362, Seattle, United States. Association for Computational Linguistics.

Linlu Qiu, Peter Shaw, Panupong Pasupat, Tianze Shi, Jonathan Herzig, Emily Pitler, Fei Sha, and Kristina Toutanova. 2022b. Evaluating the impact of model scale for compositional generalization in semantic parsing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9157–9179, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ariadna Quattoni and Xavier Carreras. 2019. Interpolated spectral NGram language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5926–5930, Florence, Italy. Association for Computational Linguistics.

Ariadna Quattoni, Xavier Carreras, and Matthias Gallé. 2017. A Maximum Matching Algorithm for Basis Selection in Spectral Learning. In *Proceedings of* the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1477–1485. PMLR.

Guillaume Rabusseau, Tianyu Li, and Doina Precup. 2019. Connecting weighted automata and recurrent neural networks through spectral learning. In *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 1630–1639. PMLR.

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

Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 922–938, Online. Association for Computational Linguistics.

Tianze Shi, Chen Zhao, Jordan Boyd-Graber, Hal Daumé III, and Lillian Lee. 2020. On the potential of lexico-logical alignments for semantic parsing to SQL queries. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1849–1864, Online. Association for Computational Linguistics.

Runxin Sun, Shizhu He, Chong Zhu, Yaohan He, Jinlong Li, Jun Zhao, and Kang Liu. 2022. Leveraging explicit lexico-logical alignments in text-to-SQL parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 283–289, Dublin, Ireland. Association for Computational Linguistics.

Ke Tran and Ming Tan. 2020. Generating synthetic data for task-oriented semantic parsing with hierarchical representations. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 17–21, Online. Association for Computational Linguistics.

Bailin Wang, Mirella Lapata, and Ivan Titov. 2021a. Structured reordering for modeling latent alignments in sequence transduction. In *Advances in Neural Information Processing Systems*, volume 34, pages 13378–13391. Curran Associates, Inc.

Bailin Wang, Wenpeng Yin, Xi Victoria Lin, and Caiming Xiong. 2021b. Learning to synthesize data for semantic parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2760–2766, Online. Association for Computational Linguistics.

Pia Weißenhorn, Lucia Donatelli, and Alexander Koller. 2022. Compositional generalization with a broad-coverage semantic parser. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 44–54, Seattle, Washington. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jingfeng Yang, Le Zhang, and Diyi Yang. 2022. SUBS: Subtree substitution for compositional semantic parsing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 169–174, Seattle, United States. Association for Computational Linguistics.

Yuekun Yao and Alexander Koller. 2022. Structural generalization is hard for sequence-to-sequence models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5048–5062, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiaoyu Yin, Dagmar Gromann, and Sebastian Rudolph. 2021. Neural machine translating from natural language to sparql. *Future Generation Computer Systems*, 117:510–519.

Yongjing Yin, Jiali Zeng, Yafu Li, Fandong Meng, Jie Zhou, and Yue Zhang. 2023. Consistency regularization training for compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1294–1308, Toronto, Canada. Association for Computational Linguistics.

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.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. AMR parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94, Florence, Italy. Association for Computational Linguistics.

Hao Zheng and Mirella Lapata. 2021. Compositional generalization via semantic tagging. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1022–1032, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hao Zheng and Mirella Lapata. 2022. Disentangled sequence to sequence learning for compositional generalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4256–4268, Dublin, Ireland. Association for Computational Linguistics.

Jing Zheng, Jyh-Herng Chow, Zhongnan Shen, and Peng Xu. 2023. Grammar-based decoding for improved compositional generalization in semantic parsing. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1399–1418, Toronto, Canada. Association for Computational Linguistics.

Victor Zhong, Mike Lewis, Sida I. Wang, and Luke Zettlemoyer. 2020. Grounded adaptation for zero-shot executable semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6869–6882, Online. Association for Computational Linguistics.

A ATISALIGNED

ATISALIGNED is an extension of the popular ATIS benchmark (Hemphill et al., 1990), in its semantic parsing version with FunQL MRs. In ATISALIGNED, 5410 NL and MR pairs are annotated by a team of four annotators for word alignments. Two annotators labeled the entire dataset, while the other two labeled a subset of 100 examples each, in order to examine the level of agreement of the labels.

Annotators were provided with an initial alignment, which was automatically generated using IBM Model 5 (Brown et al., 1993), displayed as bi-symbols of NL and MR tokens. They were then tasked with correcting the alignment. On average, annotators reported having to correct around 80% of the alignments. However, most of the corrections were minor and generally involved at most 4 simple swaps per example, which resulted in a faster annotation process compared to annotating alignments from scratch. We also found that annotators displayed a high level of agreement in the choice of head words. Disagreements were resolved by taking the majority vote among annotators.

In terms of the type of word alignments we obtained, we found that just over 9% of the examples are monotonic in this dataset, indicating that ATISALIGNED contains more complex patterns than GEOALIGNED, which contains more monotonicity.

B Constrained reorderer

In Table 5 we present the results of experiments where we constrain the re-orderer in the augmentation process. Specifically, our constrained decoding strategy restricts the output of the re-orderer to be a permutation of the input. In this way the re-orderer can not add, substitute or delete symbols of the input MR. Note also that in doing so the filter based on the re-orderer has no effect since the output MR will always have the same number of symbols as the input. We run these experiments on the GEO dataset and leave everything else unchanged in our pipeline. The constrained decoding strategy obtains improvements only on two of the IID partitions while being inferior on all the other cases, especially in the compositional partitions. These results further justify our architectural choice of leaving the re-orderer unconstrained.

C Analysis of length constraint

In our work we set ARCHER generation to have a minimum length constraint. We chose to do this in order to bias the process to produce longer sequences. We now show in this section that this is not the reason why we obtain very good performance on the length splits. We do so by showing the results of two experiments:

- ARCHER without the usage of a minimum length constraint. Note that in this case we use the same amount of augmented samples as ARCHER with the constraint.
- The GECA and Self-Training comparison where we only keep the generated samples that pass the minimum length constraint.

We report the results in table 6. We can observe that the performance between the same method is usually not dissimilar, with the exception of GECA in the query partition where adding the length constraint seems to hamper results. Besides that, ARCHER continues to perform well even without the length constraint.

D Augmented datasets

In table 7 we present the size of the augmented dataset for each partition after the filtering has been

Model	**EN**			**IT**			**DE**		
	IID	**Q**	**LEN**	**IID**	**Q**	**LEN**	**IID**	**Q**	**LEN**
Archer	**86.42**	**82.11**	**46.31**	82.50	**82.43**	**38.33**	74.52	**72.68**	**31.19**
w. constraints	85.71	79.18	38.03	**84.04**	79.02	32.44	**74.87**	71.37	29.28

Table 5: Comparison on all partitions of the Geo dataset of Archer with unconstrained reorderer vs Archer with a constrained reorderer.

Model	**Geo**						**Atis**	
	EN		**IT**		**DE**		**EN**	
	Q	**LEN**	**Q**	**LEN**	**Q**	**LEN**	**Q**	**LEN**
Without length constraint								
GECA	**87.64**	29.16	**83.57**	30.83	65.12	22.97	61.10	27.69
Self-Training	77.07	27.37	81.46	28.21	64.38	23.93	**64.91**	26.25
Archer	84.87	43.93	82.11	**39.64**	70.89	30.71	63.66	29.15
With length constraint								
GECA	77.72	27.02	80.32	31.97	60.97	24.28	60.48	26.22
Self-Training	74.30	28.81	75.60	29.28	60.16	24.40	61.83	28.15
Archer	82.11	**46.31**	82.43	38.33	**72.68**	**31.19**	63.31	**29.79**

Table 6: caption.

Dataset	**IID**	**Q**	**LEN**
Geo EN	18332	17119	9261
Geo IT	19669	16720	11218
Geo DE	12121	16745	14330
Atis	15136	15838	9529

Table 7: Sizes of the augmented datasets after filtering has been applied.

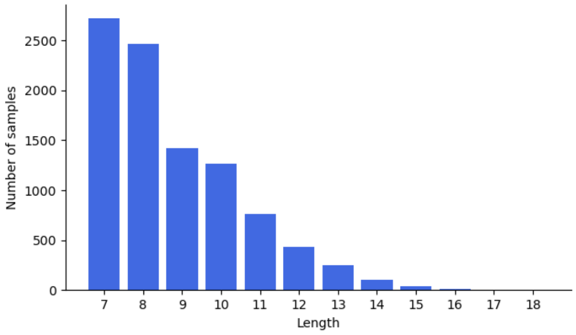

Figure 3: Distribution of the lengths of the augmented samples in the english length partition of Geo after filtering has been applied.

applied. For Geo we generate 40000 new samples and for Atis 100000.

In Figure 3 we show the distribution of the lengths of the generated samples. We consider the english length partition of Geo and generate them with a minimum length threshold of 7. The graph includes only those samples that have successfully passed the filtering step. This shows that our generation method is capable of producing longer samples to augment the dataset.

E Example of Archer generations

Table 8 reports some examples of NL and MR pairs generated by Archer. We show examples that have been labeled for correctness differently by the annotators. These include: a correct example, where both the NL and MR are deemed correct; an incorrect example; one where the MR is correct, but the NL is not; and one where the NL and MR are both correct, but the result is nonsensical according to the semantics of the sequences. Additionally, we show a correct example that showcases the ability of Archer to generate longer sequences with accurate recursions.

Correct generation
NL: what's the largest of the cities which are in maine
MR: answer(largest(city(loc_2(stateid(maine)))))
Incorrect generation
NL: what capital is the population of texas by state
MR: answer(capital(population_1(stateid(texas))))
Correct MR and incorrect NL
NL: what state has the highest population average urban population density
MR: answer(largest_one(density_1(state(all))))
Correct except semantically
NL: what is the biggest state in the state of nevada
MR: answer(largest(state(loc_2(stateid(nevada)))))
Correct recursion
NL: what states border states that border the state that borders utah
MR: answer(state(next_to_2(state(next_to_2(state(next_to_2(stateid(utah))))))))

Table 8: Examples of ARCHER generations.

F Computational details

mBART has around 610 million parameters while the WFA around 30 million (could be less depending on number of states). We run our experiments on Nvidia V100 gpus for an estimated total time of 1000 hours. The WFA was instead run on cpu. For mBART we employ the implementation of the HuggingFace library (Wolf et al., 2020), specifically *facebook/mbart-large-50*. We validate hyperparameters on the development set, usually the best configuration consists in 25 epochs, a batch size of 4 and a learning rate of $5e^{-5}$. All the results reported are the average of multiple runs.