Structurally Diverse Sampling for Sample-Efficient Training and Comprehensive Evaluation

Shivanshu Gupta[1] **Sameer Singh**[1,2] **Matt Gardner**[3]
[1]University of California Irvine [2]Allen Institute for AI [3]Microsoft Semantic Machines
{shivag5,sameer}@uci.edu, mattgardner@microsoft.com

Abstract

A growing body of research has demonstrated the inability of NLP models to generalize compositionally and has tried to alleviate it through specialized architectures, training schemes, and data augmentation, among other approaches. In this work, we study a different approach: training on instances with diverse structures. We propose a model-agnostic algorithm for subsampling such sets of instances from a labeled instance pool with structured outputs. Evaluating on both compositional template splits and traditional IID splits of 5 semantic parsing datasets of varying complexity, we show that *structurally diverse training* using our algorithm leads to comparable or better generalization than prior algorithms in 9 out of 10 dataset-split type pairs. In general, we find structural diversity to consistently improve sample efficiency compared to random train sets. Moreover, we show that structurally diverse sampling yields comprehensive test sets that are a lot more challenging than IID test sets. Finally, we provide two explanations for improved generalization from diverse train sets: 1) improved coverage of output substructures, and 2) a reduction in spurious correlations between these substructures.

1 Introduction

Systematic compositionality—expressing novel complex concepts as systematic compositions of expressions for simpler concepts—is the property underlying human languages' expressive power (Lake et al., 2017; Fodor and Pylyshyn, 1988). However, NLP models struggle to generalize to novel composite expressions in the context of semantic parsing (Lake and Baroni, 2018; Loula et al., 2018; Kim and Linzen, 2020).

Data augmentation has been extensively explored for compositional generalization (Akyürek et al., 2021; Guo et al., 2021; Wang et al., 2021; Guo et al., 2020; Qiu et al., 2022). However, instances in semantic parsing possess structure, such as abstract syntax trees (ASTs) of target programs (Figure 1). Randomly selecting instances from a grammar, or even from human annotators, while ignoring similarity in their structure is likely to produce skewed distributions with spurious correlations between substructures. This *task ambiguity* and *underspecification* are known to cause overparameterized models to have high variance and poor generalization (D'Amour et al., 2020; Geirhos et al., 2020). In contrast, a structurally diverse set of instances containing a variety of structures should better cover the combinatorial space of structures and yield more sample-efficient train sets. Moreover, this improved coverage should yield more comprehensive test sets than the traditional IID test sets.

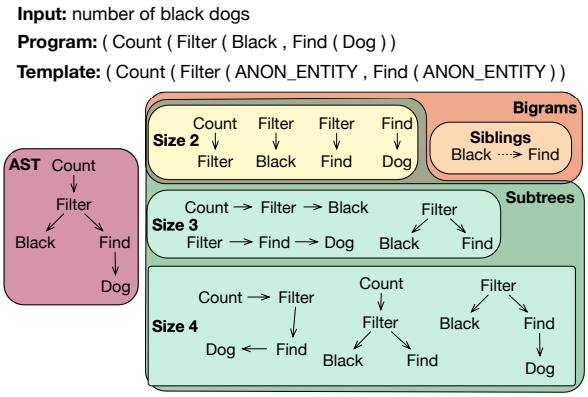

Figure 1: An instance from COVR and its abstract syntax tree (AST) and different types of substructures in it, including templates, bigrams, and subtrees. Note that subtrees will also include each node in the AST.

Prior work on diverse sampling has shown improved sample efficiency in compositional (non-IID) splits, by selecting instances to have diverse *program templates* (Oren et al., 2021) or *AST bigrams* (Bogin et al., 2022) (see Figure 1 for examples). However, for reasonably-sized programs, templates are very large structures, and bigrams very small; focusing on either does not necessarily improve diversity across a range of differently-

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4995 - 5008
December 7-11, 2022

sized structures. Moreover, there is little evidence of sample efficiency of diverse train sets in the IID setting and no exploration of diverse test sets.

In this study, we take a broader look at the advantages of structural diversity. We propose a general, substructure, and model-agnostic recipe for structurally diverse subsampling algorithms that interleaves selecting unseen substructures with selecting instances containing those substructures. Experimenting with different substructure and instance selection criteria, we show improved performance from 1) prioritizing the selection of more frequent unseen substructures in the training pool, 2) using subtrees[1] as substructures that are neither too granular (bigrams) nor too large (templates), and 3) simultaneously diversifying over both small (subtree) and larger (template) substructures.

We evaluate on template and IID splits of five semantic parsing datasets of varying complexities and find that training on diverse train sets consistently outperforms random train sets. In particular, our proposed subtree diversity algorithm is the most consistent, performing comparably with or better than the template and bigram diversity in 9 out of 10 splits, with both bigram and template algorithms often performing worse than random subsampling in IID splits. We study the efficacy of diversely sampled test sets and show that our diverse sampling algorithm yields test sets that are harder and more comprehensive than IID tests, especially with IID training sets. Finally, by comparing random and structurally diverse subsamples, we show that the latter 1) better cover the space of substructures, especially those in the long tail, and 2) have weaker spurious correlations between substructures, explaining their benefits in training and evaluation.

To conclude, our results demonstrate the effectiveness of structurally diverse train sets in inducing generalization and of structurally diverse test sets as comprehensive evaluations. We hope these insights will encourage the use of structural diversity as a criterion when sampling train or test sets from a pool of labeled instances. A setting where we expect our algorithm will be particularly useful is that of prototyping semantic parsers in zero-data settings, where a pool of programs is sampled from a grammar and mapped to canonical utterances (Wang et al., 2015; Herzig and Berant, 2019; Campagna et al., 2020; Yin et al., 2022). Another potential application could be in exemplar selection for in-context learning. Our code, data, and models are available at https://github.com/Shivanshu-Gupta/structural-diversity.

[1]These could also be called subgraphs but we'll stick to "subtree" as the whole structures are trees.

2 Setup

Semantic Parsing, the task we focus on, involves parsing an utterance x into a program or logical form y. Following Oren et al. (2021) and Bogin et al. (2022), we will assume access to a pool of instances $D_{pool} = \{e_i, \ldots, e_n\}$, where $e_i = (x_i, y_i)$, from which we wish to sample instances. As the example in Figure 1 shows, logical forms possess hierarchical structure which can be represented as trees called *abstract syntax trees* (ASTs). These structures are composed of smaller substructures such as subtrees of the AST, that are shared across logical forms. We can thus view each instance as a bag of output substructures: $\mathcal{C}(e_i) = \mathcal{C}(x_i, y_i) = \mathcal{C}(y_i) = \{c_1^{\{i\}}, \ldots, c_m^{\{i\}}\} \subseteq C$, where $\mathcal{C}$ is the mapping from instances to their substructures and C is the set of all unique substructures.

The goal of structurally-diverse sampling is to select instances that contain among them a variety of different substructures. Our hypotheses, that we verify in §6, are that a collection of such instances would both better cover the combinatorial space of structures and reduce spurious correlations between substructures compared to random instances. A structurally diverse train set should thus give the model more information about the *system* underlying the instances' structure and hence improve generalization. Similarly, its improved coverage should also make a structurally diverse test set more comprehensive. This is analogous to how one would choose to evaluate on a test set with balanced classes even when the training data might have a considerable imbalance.

3 Structurally Diverse Subsampling

3.1 Substructures

Prior work (Oren et al., 2021; Bogin et al., 2022) has used program templates and bigrams in program ASTs (Figure 1) for their diverse subsampling algorithms. Bigrams are pairs of parent-child and sibling nodes in program ASTs. Program templates represent the reasoning pattern in the program and are obtained by replacing certain program tokens such as strings and numbers with their abstract type.

Their exact implementation depends on the dataset and is described in §4.2.

While we follow Oren et al. (2021) and Bogin et al. (2022) in extracting substructures from programs alone, we choose to use different substructures, as templates and bigrams do not seem to hve the optimal granularity to diversify over. Templates are too coarse; they still share a lot of structure, and there may be a combinatorially large number of them. On the other hand, bigrams are too fine and may be unable to capture many salient structural patterns. We thus use subtrees of the program AST up to a size d. The ASTs are constructed as in Bogin et al. (2022): tokens in a program are categorized as either functions, values, or structural tokens (such as parentheses or commas) that define the hierarchical structure of the program.

3.2 Algorithm

As motivated in §2, we want to sample instances containing among them a variety of substructures. Additionally, we wish to experiment with: 1) prioritizing substructures more common in the pool, as we expect it to reduce the divergence of the subsample from the pool distribution; and 2) simultaneously diversifying over both fine (subtree) and coarse (template) granularity substructures. One approach would be constructing an optimization problem that directly selects an optimally diverse set of instances. However, this has the challenge of defining a measure of diversity that is also tractable to optimize for large pools and a large number of substructures (see Table 2). Additonally, it would be harder to experiment with the different variations of diverse sampling described above.

We thus take the route of an iterative algorithm (pseudo-code in Algorithm 1) that alternates between picking a substructure and picking an instance with that substructure till the requisite number, B, of instances has been sampled. Here, w_c and w_e specify the substructure and instance selection criteria by assigning weights based on the current state comprising sampled instances D_{sample}, substructures C_{sample}, and templates T_{sample}. The algorithm resets C_{sample} and T_{sample} once all substructures and templates, respectively have been sampled to allow cycling over them repeatedly. In the following sections, we will show that with different substructure definitions and substructure and instance-weighting schemes, Algorithm 1 can subsume both the template diversity and bigram diversity algorithms from Oren et al. (2021) and Bogin et al. (2022). Additionally, it will allow us to experiment with the variations described above.

Algorithm 1 Structurally Diverse Subsampling

Require: Instance pool D_{pool}; set of all substructures C, and templates T in pool: instance-to-substructure mapping $\mathcal{C}$; template mapping $\mathcal{T}$; substructure weight function w_c; instance weight function w_e; training budget B

$D_{sample}, C_{sample}, T_{sample} \leftarrow \phi$
$i \leftarrow 0$
while $i < B$ **do**
  $c = \underset{c \in C}{\text{argmax}}\, w_c(c, D_{pool}, D_{sample}, C_{sample})$
  $e = \underset{\substack{e \in D_{pool} \\ \text{s.t. } c \in \mathcal{C}(e)}}{\text{argmax}}\, w_e(e, D_{pool}, D_{sample}, T_{sample})$
  $D_{pool} \leftarrow D_{pool} \setminus e$
  $D_{sample} \leftarrow D_{sample} \cup e$
  $C_{sample} \leftarrow C_{sample} \cup c$
  $T_{sample} \leftarrow T_{sample} \cup \{\mathcal{T}(e)\}$
  $C \leftarrow \bigcup_{e \in D_{pool}} \mathcal{C}(e)$
  $T \leftarrow \{\mathcal{T}(e) : e \in D_{pool}\}$
  if $C_{sample} = C$ **then**
    $C_{sample} \leftarrow \phi$
  end if
  if $T_{sample} = T$ **then**
    $T_{sample} \leftarrow \phi$
  end if
end while
return D_{sample}

3.3 Subtree Diversity

Our proposed subsampling algorithm diversifies over subtrees as defined in §3.1, i.e., $\mathcal{C}(x, y)$ is the set of subtrees of size $\leq d$ in the AST of y. We will use $d = 4$. Since we want to prioritize selecting more frequent unsampled substructures in the pool, we will use $w_c(c) = \mathbb{1}[c \notin C_{sample}]F_{pool}(c)$ where $F_{pool}(c)$ is the number of instances in the pool containing c.

For selecting instances given a sampled substructure, we use these instance weighting schemes:

1. RANDEX samples an instance uniformly at random: $w_e(e)$ is a constant.
2. RANDNEWT samples an instance with unseen template: $w_e(e) = \mathbb{1}[\mathcal{T}(e) \notin T_{sample}]$.
3. FREQNEWT samples instance with the most frequent unsampled template: $w_e(e) = \mathbb{1}[\mathcal{T}(e) \notin T_{sample}]F_{pool}(\mathcal{T}(e))$ where $\mathcal{T}(e)$ is e's template and $F_{pool}(t)$ is the number of instances in the pool with template t.

The last two schemes seek to improve the coverage of templates to allow the diversifying of both subtrees and templates.

3.4 Template and Bigram Diversity

Template Diversity from Oren et al. (2021), henceforth referred to as TEMPLATE, can be implemented in the framework of Algorithm 1 as:

- $\mathcal{C}(x, y)$ is a singleton set containing the template for y.
- $w_c(c)$ is a constant function i.e. a random template.
- $w_e(e)$ is a constant function (RANDEX).

Additionally, we also experimented with selecting only among the unsampled templates i.e. $w_c(c) = \mathbb{1}\left[c \notin C_{sample}\right]$ and with prioritizing more frequent unsampled templates using the substructure weighting scheme of §3.3. We found both of these to improve performance but only include the results for the latter. We will refer to it as TEMPLATE[FREQ].

Bigram Diversity from Bogin et al. (2022) , hereafter referred to as BIGRAM, can also be implemented in the framework of Algorithm 1 as:

- $\mathcal{C}(x, y)$ is the set of bigrams in y's AST as defined by Bogin et al. (2022).
- Bogin et al. (2022)'s bigram diversity algorithm randomly samples from unsampled bigrams until there is still an unsampled bigram and then any random bigram. This can be formulated as: $w_c(c) = \mathbb{1}\left[c \notin \tilde{C}_{sample}\right]$ where $\tilde{C}_{sample} = \bigcup_{e \in D_{sample}} \mathcal{C}(e)$ is the set of all bigrams in the current sample.
- $w_e(e)$ is a constant function (RANDEX).

We also experimented with bigram diversity with the substructure weighting scheme of §3.3. We will refer to this algorithm as BIGRAM[FREQ].

4 Experiments

Given a dataset $\mathcal{D}$ of utterance-program pairs, we create three different types of splits with each split consisting of a training pool D_{pool} and test set D_{test}. We then compare the various subsampling algorithms described in §3 by using them to sample training sets D_{train} of varying budget size B from D_{pool} and evaluating on D_{test}.

4.1 Splits

IID split For this we randomly sample instances from $\mathcal{D}$ to use as D_{test}, keeping the rest for D_{pool}.

Template split This is a type of compositional split proposed by (Finegan-Dollak et al., 2018). Here instances are grouped based on their program template as described in §3.1 The split is then created by randomly splitting the set of templates into a train set and a test set and using examples for train or test templates as D_{pool} or D_{test} respectively. We follow the procedure of Bogin et al. (2022) to obtain solvable template splits where every token in the test set also occurs in the train set.

Subtree split In §2 we argued that diversely subsampled sets of instances should also make for more comprehensive test sets. We thus experiment with a third type of split: we use SUBTREE[FREQNEWT] diverse subsampling to sample test sets D_{test} from $\mathcal{D}$, keeping the rest as D_{pool}.

4.2 Datasets

We use five semantic parsing datasets from diverse domains and complexity for our analysis, with both synthetic and natural language input utterances. Tables 1 and 2 show a few examples and statistics regarding number of instances and different types of substructures.

COVR: A synthetic dataset that uses a variable-free functional query language and is generated using a synchronous context-free grammar (SCFG) adapted from the VQA dataset of Bogin et al. (2021a). We use the SCFG to generate 100K examples for our experiments.

ATIS (Hemphill et al., 1990; Dahl et al., 1994): A dataset of natural language queries about aviation paired with λ-calculus programs.

Overnight (Wang et al., 2015): A dataset containing both synthetic and natural language utterances from 11 domains (e.g. *socialnetwork, restaurants*, etc.) paired with Lambda-DCS logical forms.

Schema2QA (Xu et al., 2020): Uses the ThingTalk language (Campagna et al., 2019). We use the synthetic instances from the *people* domain generated by Oren et al. (2021).

SM-CalFlow (Andreas et al., 2020): Consists of dialogs paired with LISP programs. Each instance is a single dialogue turn from one of two domains about creating calendar events or querying an org chart.

Except for SM-CalFlow, which we took from Andreas et al. (2020), we used the preprocessed versions of the above datasets provided by Bogin et al. (2022) and used their code[2] to anonymize programs and produce ASTs. For SM-CalFlow, we anonymized strings and numbers such as "staff

[2] https://github.com/benbogin/unobserved-local-structures

Dataset	Input Utterance	Target Program
COVR (synthetic)	*What is the number of black dog that is chasing mouse ?*	count (with_relation (filter (black , find (dog)) , chasing , find (mouse)))
OVERNIGHT (synthetic†, natural°)	†*person whose height is 180 cm and whose birthdate is 2004* °*what person born in 2004 is 180 cm tall*	(listValue (filter (filter (getProperty (singleton en.person) (string !type)) (string height) (string =) (number 180 en.cm)) (string birthdate) (string =) (date 2004 -1 -1)))
SCHEMA2QA (synthetic)	*what people are named aideliz li*	(Person) filter id = "aideliz li"
ATIS (natural)	*a flight on continental airlines leaving boston and going to denver*	(lambda $0 e (and (flight $0) (airline $0 co : al) (from $0 boston : ci) (to $0 denver : ci)))
SM-CALFLOW (natural)	*When is my next staff meeting scheduled for?*	(Yield (Event.start (FindNumNextEvent (Event.subject? (? = "staff meeting")) 1L)))

Table 1: Examples of input utterance and target program pairs for the datasets used in this work.

Dataset	Instances	Bigrams	Subtrees	Templates
ATIS	5037	5091	41536	1149
COVR	100000	298	4490	29141
OVERNIGHT	4419	354	3015	87
SM-CALFLOW	106072	43689	208527	21082
SCHEMA2QA	1577860	223	3060	139

Table 2: Number of instances, bigrams, subtrees (size ≤ 4), and templates in the datasets used for structural diversity experiments.

meeting" and "1L" in Table 1 and used nltk[3] to produce ASTs. Additionally, we excluded all instances with a particular program that accounted for more than 12% of the original dataset.

4.3 Model and Training

We use the pre-trained BART-base model (Lewis et al., 2020) for our experiments, fine-tuning it on subsamples for each dataset, split, and subsampling algorithm. For each dataset, we use 4 different random seeds to create 4 splits of each type. Then for each split and training budget, we use 3 different seeds to subsample 3 training sets for each subsampling algorithm. See App. A for more details.

5 Results

5.1 Structurally Diverse Train Sets

Figure 2 compares the various structurally diverse sampling algorithms, and random sampling on Template and the IID splits of the different datasets.

Structural Diversity improves sample efficiency. Random sampling is outperformed by structurally diverse sampling in 9 out of 10 dataset-split type combinations with about 2x sample efficiency in template splits of all five datasets except COVR and IID splits of SCHEMA2QA and OVERNIGHT. The only exception is the IID split of SM-CALFLOW. We believe this is due to a greater imbalance in substructures in SM-CALFLOW where over a third of templates (as well as bigrams and subtrees) only appear in a single instance.

Subtree diversity is most consistent. Our proposed subtree diversity algorithm is the most consistent among the various diverse sampling algorithms. In all but 1 of the 10 splits, it outperforms or matches both template and bigram diversity which often lag behind even random sampling. The only exception is template diversity in the template split of SCHEMA2QA, which has very few templates (see Table 2). These results suggest that the optimal substructure granularity depends on the task. For datasets with little structural diversity, diversifying over templates is sufficient. However, diversifying over smaller substructures is more beneficial for complex datasets with many program templates overlapping in structure as it exploits the structural similarity of programs.

Combining subtree and template diversity improves performance. Comparing SUBTREE[RANDNEWT] and SUBTREE[RANDEX] we see that the former performs better in template splits with minor reductions in IID splits. This suggests that simultaneously diversifying over fine and coarse substructures, subtrees, and templates here, can be more effective than either alone.

Prioritizing more frequent compounds improves efficiency. Figure 6 compares BIGRAM and TEMPLATE with BIGRAM[FREQ] and TEM-

[3] https://www.nltk.org/

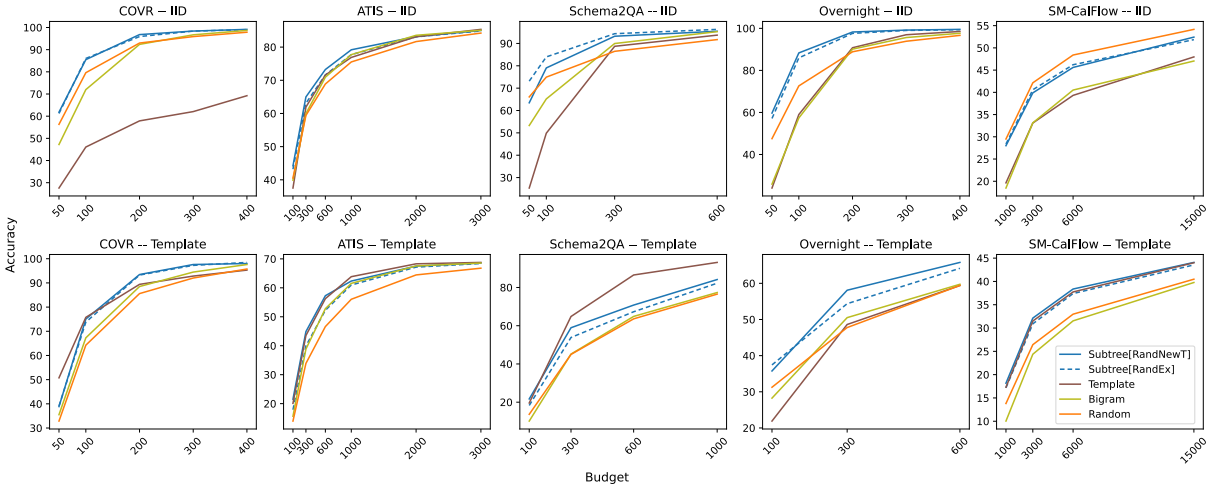

Figure 2: Comparing different subsampling algorithms on IID and Template splits for varying budgets.

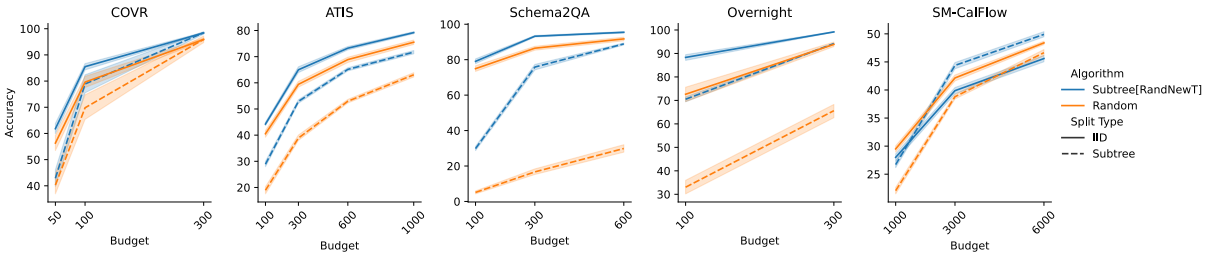

Figure 3: Subtree splits are more comprehensive than IID splits and hence are much harder with randomly sampled train sets but not with diverse train sets.

PLATE[FREQ] respectively. It is evident that on IID splits, bigram and template diversity greatly benefit from prioritizing more frequent substructures instead of random selection, with only minor degradation in template splits.

5.2 Structurally Diverse Test Sets

Figure 3 compares the performance of random and diverse (SUBTREE[RANDEX]) subsamples in IID and Subtree splits. It is evident that Subtree splits are more challenging than IID splits. More importantly, this split is much harder with random train sets than with diverse train sets, which, as the training budget increases, quickly close the gap with IID performance on all datasets except ATIS. This is expected since, unlike template splits, this split is challenging not because of a systematic distributional gap between train and test sets, but because it tests for a lot more structures that a random train set may not cover. We thus believe that structurally diverse test sets also enable more comprehensive evaluation.

6 Analysis

Having seen that structurally diverse datasets improve generalization, we now investigate why they do so. In §2 we hypothesize that structural diversity has two benefits over random sampling: (1) improved coverage of the space of substructures, and (2) reduced spurious correlations. We now evaluate whether this is indeed the case.

We will view each instance as the set of subtrees contained in it, i.e., $e_i \subset C$ where C is the set of all subtrees in the dataset. We define the frequency of a subtree, s, in a set of instances A as the number of instances containing it, i.e. $F_A(s) = |\{s \in \mathbf{e} : \mathbf{e} \in A\}|$. The frequency of a pair of subtrees, s_i, s_j, is analogously defined as $F_A(s_i, s_j) = |\{s_i, s_j \in \mathbf{e} : \mathbf{e} \in A\}|$. The rank of a subtree in A is defined as its position in a list of all subtrees descending in frequency, with equally-frequent subtrees assigned the average of their ranks.

6.1 Improved Coverage of the Long tail

To verify whether structural diversity improves the coverage of substructures, we use random and SUBTREE[RANDEX] subsampling to take equal-sized subsamples from pools consisting of entire datasets, $\mathcal{D}$, and compare the distribution of subtrees. Figure 4 shows the number of unique subtrees in random and diverse subsamples bucketed by rank in $\mathcal{D}$. As can be seen in the figure, structurally diverse

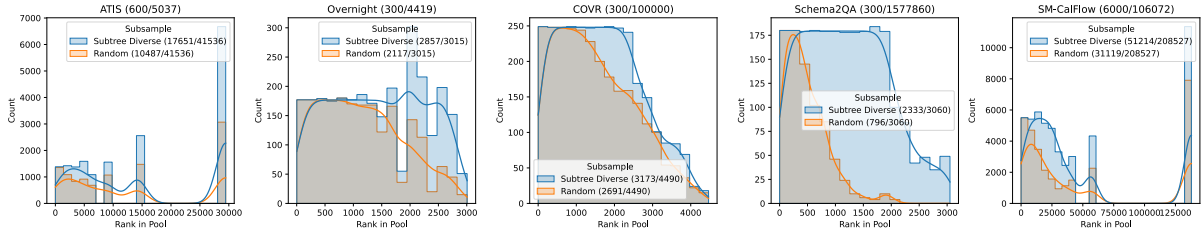

Figure 4: The number of unique subtrees in diverse and random subsamples bucketed by the rank of their frequency in the pool. Titles include the size of the subsamples and the pool, and the legends include the number of unique subtrees in the subsamples v/s the pool.

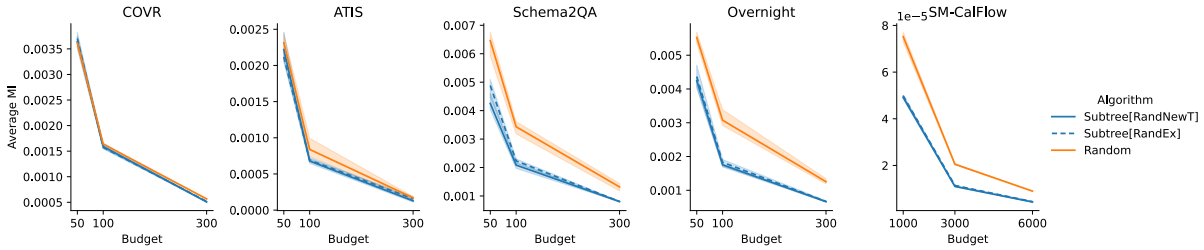

Figure 5: Average Mutual Information (AMI) of random subsamples compared with that of diverse subsamples.

sampling improves the coverage of substructures, especially those in the long tail. Crucially, because substructures are chosen by frequency, this coverage of the long tail does not come at the expense of frequent substructures. We refer the reader to Appendix C for additional analyses showing that although SUBTREE[RANDEX] sampling directly diversifies over only output subtrees, it also improves coverage of program templates as well as n-grams in input utterances.

6.2 Weakened Spurious Correlations

Structurally diverse datasets might improve generalization by weakening the spurious correlations between substructures that would exist in any dataset of bounded size.

Measure for Correlations We measure correlations between substructures in a set of instances D as the sum of pairwise mutual information (MI) between them, normalized to account for the larger number of distinct substructures in diverse subsamples. Let $S = s_1, \ldots, s_m$ be the set of all subtrees in a sample. We define an indicator random variable I_s for each subtree s that is 1 if s is present in a random instance, 0 otherwise. MI between two subtrees, s_i and s_j is then defined as,

$$\mathrm{MI}[s_i, s_j] = -\sum_{I_{s_i}, I_{s_j}} \tilde{p}(I_{s_i}, I_{s_j}) \log \frac{\tilde{p}(I_{s_i}, I_{s_j})}{\tilde{p}(I_{s_i})\tilde{p}(I_{s_j})} \quad (1)$$

where $\tilde{p}(I_s = 1)$ and $\tilde{p}(I_{s_i} = 1, I_{s_j} = 1)$ are empirical probabilities defined as,

$$\tilde{p}(s = 1) = \frac{|\{s \in \mathbf{e} : \mathbf{e} \in D\}|}{|D|} \quad (2)$$

$$\tilde{p}(s_i = 1, s_j = 1) = \frac{|\{s_i, s_j \in \mathbf{e} : \mathbf{e} \in D\}|}{|D|} \quad (3)$$

Other probabilities are defined analogously. Finally, our measure of correlations is the average mutual information (AMI) across all subtree pairs:

$$\mathrm{AMI}(D) = \frac{1}{|S|^2} \sum_{i,j} \mathrm{MI}[s_i, s_j] \quad (4)$$

Split Type	Budget	COVR	Overnight	Schema2QA
IID	100	-0.50	-0.83	-0.39
	300	-0.70	-0.76	-0.56
Template	100	-0.57	-0.58	-0.68
	300	-0.60	-0.56	-0.64

Table 3: Spearman correlations of AMI and Accuracy for train sets sampled randomly and using SUBTREE[RANDEX] and SUBTREE[RANDNEWT] algorithms averaged across 4 split seeds (from § 4).

Results For each dataset, we took 3 samples using random and subtree subsampling algorithms, treating the entire dataset as pool, and computed the AMI over their substructures. Figure 5 shows that diverse subsamples have lower AMI than random subsamples across different training budgets and the different split types, confirming our hypothesis that structurally diverse datasets have lower correlation. Moreover, examining Spearman correlations

of AMI and accuracy (Table 3) for train sets subsampled using random, SUBTREE[RANDEX] and SUBTREE[RANDNEWT] subsampling algorithms from the experiments in §4, we find that they are significantly negatively correlated, giving further credence to our measure. These results substantiate our hypothesis that reducing spurious correlations is indeed one way diverse subsamples improve generalization.

7 Related Work

Generalization Work on improving compositional generalization has considered many approaches including specialized architectures (Herzig and Berant, 2021; Bogin et al., 2021b; Chen et al., 2020; Gordon et al., 2020; Yin et al., 2021), data augmentation (Andreas, 2020; Akyürek et al., 2021; Guo et al., 2021), modifications to training methodology (Oren et al., 2020; Csordás et al., 2021), and meta learning (Conklin et al., 2021; Lake, 2019). Of these, data-augmentation has the advantage of being model-agnostic; however, as argued in this work, randomly selecting training instances is inefficient. Our work builds upon Oren et al. (2021) and Bogin et al. (2022) in arguing for structurally diverse training.

Training Many methods have been explored for training instance selection, including: 1. Active Learning (AL) (Lewis and Catlett, 1994; Settles and Craven, 2008), where instances are iteratively selected for annotation based on criteria such as model uncertainty (Gal et al., 2017) or diversity (Sener and Savarese, 2018); and 2. adversarial selection of instances that model fails on (Bras et al., 2020; Sakaguchi et al., 2020; Wallace et al., 2019; Nie et al., 2020; Kiela et al., 2021). However, given that these methods use a model in the loop for selection and generally work in the input domain, they are not comparable to our method. Additionally, the coupling of the dataset and model, and the lack of generalizability of AL heuristics has been shown to limit the effectiveness of active learning in practice (Lowell et al., 2019; Karamcheti et al., 2021). Tamkin et al. (2022) recently showed that pretrained models benefit more from AL by preferring ambiguous instances or ones with uncommon feature(s) thereby reducing task under-specification which can lead to higher variance and instability in models under-constrained by their training datasets (D'Amour et al., 2020; Geirhos et al., 2020). Our analyses from §6 show that structurally diverse sampling also attempts to reduce under-specification albeit by directly sampling from a pool model-agnostically.

Evaluation Out-of-distribution (OOD) tests are essectial as testing in-distribution may not penalize models for learning spurious patterns (Linzen, 2020). Our work on sampling diverse test sets is most related to methods for creating splits to test for compositional generalization from existing datasets (Keysers et al., 2020; Shaw et al., 2021; Bogin et al., 2022). However, unlike these, diverse test sets do not attempt to create a systematic gap between the train and test sets, but, as shown in §6.1, cover the space of structures better.

8 Conclusion

In this work, we studied the benefits of structural diversity for the task of semantic parsing. We proposed a novel model-agnostic algorithm for sampling structurally diverse instances by diversifying over subtrees in program ASTs. Evaluating on multiple datasets with varying complexities and on both IID and compositional splits, we demonstrated that diversity almost always, and often significantly, improves generalization. We further demonstrated that structural diversity also yields more comprehensive test sets than traditional IID test sets. Finally, we showed that these benefits of structural diversity are likely a manifestation of improved coverage of the long-tail of substructures as well as a reduction in spurious correlations between them.

We hope that our results demonstrating the importance of structural diversity will encourage future research on better structurally diverse sampling algorithms as well as their use to guide train and test set construction. The algorithms in this work are applicable whenever a large pool of possible output programs is available. One such setting is of prototyping a semantic parser in a zero-data setting where a pool of outputs can be sampled from a grammar and mapped to canonical utterances that can subsequently be paraphrased for linguistic variation either manually or automatically (Wang et al., 2015; Campagna et al., 2020). Another scenario is that of selecting instances for in-context or few-shot learning (Brown et al., 2020).

Perhaps more importantly, our results indicate that the failure of NLP models on compositional generalization may be due in part to the lack of diversity in them, suggesting the need to create more diverse benchmarks.

Acknowledgements

We would like to thank the anonymous reviewers for their feedback. Further we'd also thank Ben Bogin, Yasaman Razeghi, and Catarina Belem for their useful comments. This work was sponsored in part by the DARPA MCS program under Contract No. N660011924033 with the United States Office Of Naval Research, in part by funding by AI2, and in part by the NSF grant #IIS-1817183. The views expressed are those of the authors and do not reflect the policy of the funding agencies.

Limitations

A key limitation of the algorithms discussed in this work is their limited applicability due to the assumption of the availability of a pool of output programs paired with input utterances (natural or canonical). To alleviate these, future work can look at diverse sampling directly from a grammar without the intermediate step of sampling a pool or by preferring structural diversity in the input space instead of the output space i.e. on natural language utterances. The latter will additionally allow these methods to be applied to linguistic tasks other than semantic parsing.

Additionally, we took the route of a greedy iterative sampling algorithm to allow us to tractably subsample from large pools with large number of substructures. Future work can thus explore tractable optimization of some measure of diversity of sets of instances.

Ethics Statement

In this work, we studied the efficacy of structurally diverse sampling when creating training and test sets for semantic parsing. Since we focus on structures derived from output programs and not language utterances it is unlikely to have any direct social impact or induce any biases in NLP systems.

References

Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. 2021. Learning to recombine and resample data for compositional generalization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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.

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. 2020. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

Ben Bogin, Shivanshu Gupta, and Jonathan Berant. 2022. Unobserved local structures make compositional generalization hard.

Ben Bogin, Shivanshu Gupta, Matt Gardner, and Jonathan Berant. 2021a. COVR: A test-bed for visually grounded compositional generalization with real images. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9824–9846, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ben Bogin, Sanjay Subramanian, Matt Gardner, and Jonathan Berant. 2021b. Latent compositional representations improve systematic generalization in grounded question answering. *Transactions of the Association for Computational Linguistics*, 9:195–210.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1078–1088. PMLR.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 122–132, Online. Association for Computational Linguistics.

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, page 394–410, New York, NY, USA. Association for Computing Machinery.

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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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*.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. 2020. Underspecification presents challenges for credibility in modern machine learning.

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.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. PMLR.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Jonathan Gordon, David Lopez-Paz, Marco Baroni, and Diane Bouchacourt. 2020. Permutation equivariant models for compositional generalization in language. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

Yinuo Guo, Hualei Zhu, Zeqi Lin, Bei Chen, Jian-Guang Lou, and Dongmei Zhang. 2021. Revisiting iterative back-translation from the perspective of compositional generalization. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 7601–7609. AAAI Press.

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. 2019. Don't paraphrase, detect! rapid and effective data collection for semantic parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3810–3820, Hong Kong, China. Association for Computational Linguistics.

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.

Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher Manning. 2021. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. 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 7265–7281, Online. Association for Computational Linguistics.

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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

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 M. Lake. 2019. Compositional generalization through meta sequence-to-sequence learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9788–9798.

Brenden M. 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, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2879–2888. PMLR.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40.

David D. Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In William W. Cohen and Haym Hirsh, editors, *Machine Learning Proceedings 1994*, pages 148–156. Morgan Kaufmann, San Francisco (CA).

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

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. 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.

David Lowell, Zachary C. Lipton, and Byron C. Wallace. 2019. Practical obstacles to deploying active learning. 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 21–30, Hong Kong, China. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

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.

Linlu Qiu, Peter Shaw, Panupong Pasupat, Pawel Nowak, Tal Linzen, Fei Sha, and Kristina Toutanova. 2022. 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.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, Honolulu, Hawaii. 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.

Alex Tamkin, Dat Nguyen, Salil Deshpande, Jesse Mu, and Noah Goodman. 2022. Active learning helps pretrained models learn the intended task.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Bailin Wang, Wenpeng Yin, Xi Victoria Lin, and Caiming Xiong. 2021. 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.

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, Beijing, China. Association for Computational Linguistics.

Silei Xu, Giovanni Campagna, Jian Li, and Monica S. Lam. 2020. Schema2qa: High-quality and low-cost q&a agents for the structured web. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 1685–1694. ACM.

Pengcheng Yin, Hao Fang, Graham Neubig, Adam Pauls, Emmanouil Antonios Platanios, Yu Su, Sam Thomson, and Jacob Andreas. 2021. Compositional generalization for neural semantic parsing via span-level supervised attention. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2810–2823, Online. Association for Computational Linguistics.

Pengcheng Yin, John Wieting, Avirup Sil, and Graham Neubig. 2022. On the ingredients of an effective zero-shot semantic parser. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1455–1474, Dublin, Ireland. Association for Computational Linguistics.

Budget	#Epochs
50	160
300	128
600	96
1000	80

Table 4: Number of training epochs for COVR, ATIS, Schema2QA and Overnight depending on train set size.

A Training

Models for COVR, ATIS, Schema2QA and Overnight datasets were trained for different number of epochs depending on train set size as shown in Table 4. Models for Schema2QA were all trained for 240 epochs. Training was run with batch sizes ranging from 8 to 20, depending on the maximum number of example tokens in each dataset and a learning rate of $3e^{-5}$ with polynomial decay. Each experiment was run with a Nvidia Titan RTX GPU and took between a few minutes to a couple of hours as we varied the training set size and number of epochs. We used exact match accuracy as our metric and following (Bogin et al., 2022), we do early stopping using the test set. As our goal is to estimate the train set quality and not the model, we argue this is an acceptable choice in our setting.

B Additional Results

The original Bigram and Template diversity algorithms (BIGRAM and TEMPLATE), while good in template splits, are worse than even random subsampling in IID splits. However, replacing their substructure selection scheme with one that prioritizes frequent substructures (BIGRAM[FREQ] and TEMPLATE[FREQ]) improves both of their efficiencies in IID splits with minor degradation in template splits (Figure 6). The only exception is the template diversity in COVR. This is expected given that it was generated from a synchronous grammar with production rules sampled uniformly at random and hence is dominated by instances with shorter templates. Thus, TEMPLATE[FREQ] will only pick these. Additionally, the slightly poorer performance subtree diversity than template diversity on SM-CALFLOW can be attributed to (1) imperfect program to tree conversion and (2) the presence of free-form strings in programs which together lead to a large number of spurious subtrees.

C Coverage

Figure 7 replicates the analysis of § 6.1 using program templates and n-grams in the input utterances as substructures showing that improving coverage over subtrees also leads to improved coverage of program templates as well as n-grams input utterances.

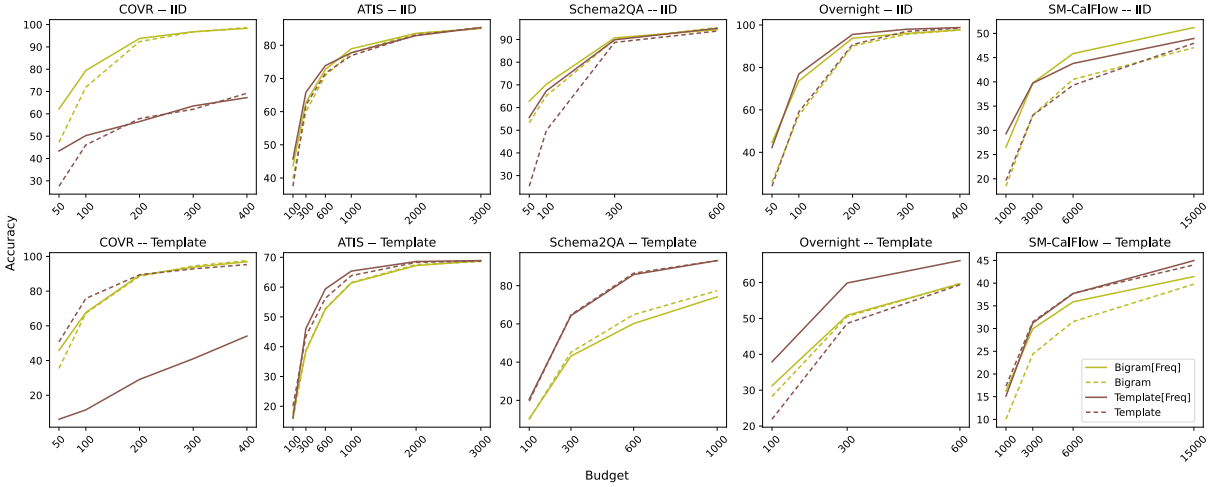

Figure 6: Prioritizing more frequent substructures greatly improves bigram and template diversity in IID splits with relatively minor degradation in template splits.

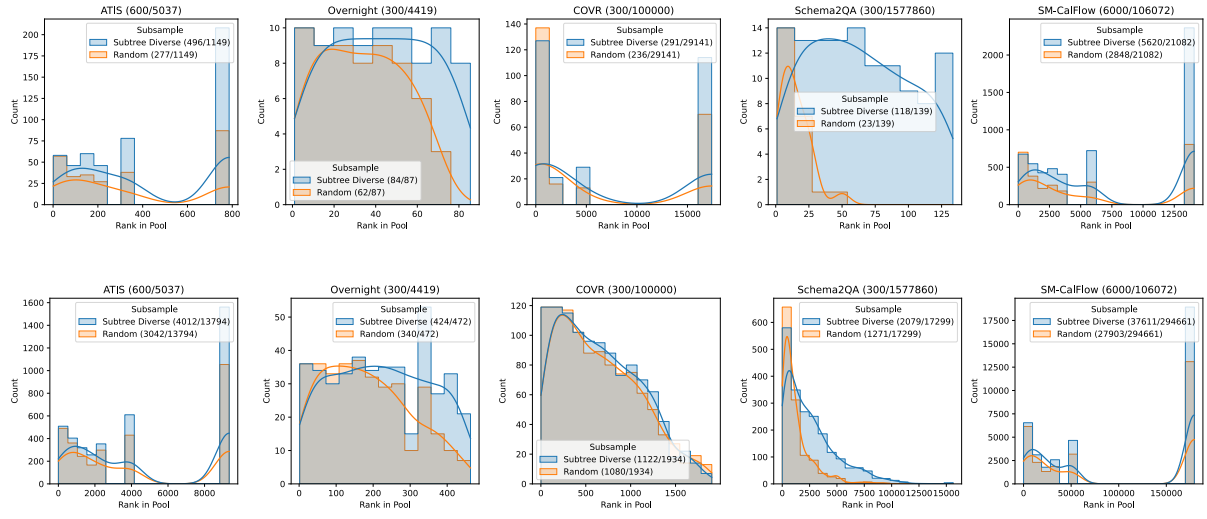

Figure 7: Number of unique program templates (top) and input utterance n-grams of size up to three (bottom) in diverse and random subsamples bucketed by pool ranks. Titles include size of the subsample v/s the pool while the legends include the number of unique substructures in each subsample v/s the pool.