# *-CFQ: Analyzing the Scalability of Machine Learning on a Compositional Task

**Dmitry Tsarkov[†‡], Tibor Tihon[†], Nathan Scales, Nikola Momchev, Danila Sinopalnikov, Nathanael Schärli**

Google Research, Brain Team

{tsar,ttihon,nkscales,nikola,sinopalnikov,schaerli}@google.com

**Abstract**

We present *-CFQ ("star-CFQ"): a suite of large-scale datasets of varying scope based on the CFQ semantic parsing benchmark, designed for principled investigation of the scalability of machine learning systems in a realistic compositional task setting. Using this suite, we conduct a series of experiments investigating the ability of Transformers to benefit from increased training size under conditions of fixed computational cost. We show that compositional generalization remains a challenge at all training sizes, and we show that increasing the scope of natural language leads to consistently higher error rates, which are only partially offset by increased training data. We further show that while additional training data from a related domain improves the accuracy in data-starved situations, this improvement is limited and diminishes as the distance from the related domain to the target domain increases.

## 1 Introduction

Intuitively, if you see a lot of examples of natural language questions about TV shows, it ought to also help understand similar syntax in questions about movies, or in questions that refer to both movies and TV shows together. Ideally, the training examples from the related domain should strictly improve performance, not hurt it. If you can satisfy that property, then you have at least a chance at eventually achieving arbitrarily robust performance across a range of domains, given sufficient training data in aggregate.

How and to what extent current machine learning (ML) approaches can be made to robustly solve natural language understanding (NLU) at the scale of arbitrary natural language across domain – with or without access to large quantities of training data – remains, however, an open question.

On one hand, research into the scaling behavior of deep learning systems has found generalization loss to decrease reliably with training size and model size in a power law or related logarithmic relationship across a range of architectures and tasks, from image classification with convolutional neural networks (Cho et al. 2015) to language modeling with Transformers (Rosenfeld et al. 2019; Kaplan et al. 2020; Brown et al. 2020). Recent results in an i.i.d. setting show this pattern to persist across many orders of magnitude, with no established upper limit (Kaplan et al. 2020).

At the same time, it has been shown that current ML systems continue to struggle to achieve robust performance in classes of tasks that require compositional generalization (Keysers et al. 2020) – an ability that has been argued to be crucial to robust language understanding (Fodor, Pylyshyn et al. 1988; Lake and Baroni 2018; Battaglia et al. 2018; Hupkes et al. 2019).

In this paper, we combine these two lines of research by investigating the effect of training size on error rates in the context of a compositional task. Specifically, we derive a suite of extended datasets based on the Compositional Freebase Questions (CFQ) semantic parsing benchmark (Keysers et al. 2020). We then use the compositional structure of each example to construct controlled experiments that measure the error rates when increasing training size in settings requiring compositional generalization and in settings simulating scaling to a broader scope of natural language. We apply these experiments to analysis of Transformers (Vaswani et al. 2017) in a setting of fixed computational cost – that is, of fixed model size and fixed training steps – and demonstrate key limits to their scalability in this setting.

Our contributions are the following:

- We present *-CFQ ("star-CFQ"): a suite of large-scale datasets and corresponding canonical splits, designed to enable principled investigation of the scalability of ML systems in a realistic compositional task setting. The datasets and splits follow the same setup as the original CFQ, but span a range of data sizes, rule scopes, and compound divergences with the largest consisting of 76% more rules, 41x as many examples, 14x as many question patterns, 53x as many SPARQL patterns, and covering a 32x larger domain than CFQ (Section 3).
- We confirm that under conditions of fixed computational cost, error rates for Transformers plateau at large training sizes. Moreover, targeting larger scopes of natural language leads to consistently higher error rates which are only partially offset by increased training data (Section 4).
- We demonstrate that compositional generalization remains a challenge at all training sizes, but that increases in training size continue to reap benefits in situations requir-

[†]Equal contribution. For author contributions, see Appendix A.

[‡]Version with appendices: http://arxiv.org/abs/2012.08266

ing compositional generalization, even when error rates would seem to have plateaued in the i.i.d. case (Section 5).

- We show that while access to additional training data from a related domain improves accuracy in data-starved situations, this improvement is limited and diminishes as the distance from the related domain to the target domain increases (Section 6).

## 2 CFQ Benchmark

Keysers et al. (2020) introduced the Compositional Freebase Questions (CFQ), which is a simple but realistic and large natural language dataset that is specifically designed to measure compositional generalization. The task of interest is semantic parsing from a natural language question (such as 'Which art director of [Stepping Sisters 1932] was a parent of [Imre Sándorházi]?') to a SPARQL query, which can then be executed against the Freebase knowledge base. Named entities are anonymized, which is standard practice and ensures that the models do not have to learn all the entities.

CFQ was constructed using the Distribution-Based Compositionality Assessment (DBCA) method. This means that the dataset is automatically generated from a set of rules in a way that precisely tracks which rules (atoms) and rule combinations (compounds) were used to generate each example. Using this information, the authors generate "maximum compound divergence" (MCD) splits, which maximize the compound divergence while guaranteeing a small atom divergence between train and test sets. MCD splits are well suited for assessing compositionality because they are both fair (because the distribution of individual rules is similar) and compositionally challenging (because the distribution of compounds is as different as possible).

The authors release a number of MCD splits for CFQ, and show that there is a strong negative correlation between the accuracy of three standard sequence-to-sequence architectures and the compound divergence. They investigate this for *LSTM+attention* (an LSTM (Hochreiter and Schmidhuber 1997) with attention mechanism (Bahdanau, Cho, and Bengio 2015)), for *Transformer* (Vaswani et al. 2017), and for *Universal Transformer* (Dehghani et al. 2018).

In a follow-up publication, Furrer et al. (2020) show that this negative correlation also applies to architectures that specifically target compositional generalization (such as CGPS (Li et al. 2019) and Neural Shuffle-Exchange Networks (Freivalds, Ozoliņš, and Šostaks 2019)) and cannot be overcome by masked language model pre-training using the Text-to-Text Transfer Transformer (T5) (Raffel et al. 2019).

## 3 *-CFQ

We present here ***-CFQ**[1], a suite of datasets building on the same overall structure and base rule set as CFQ, but with two key differences intended to facilitate investigation of the scalability of solutions to the semantic parsing task:

- **Increased data size:** The datasets span a range of data sizes, with the largest 41x the size of CFQ.

[1]Available at https://github.com/google-research/google-research/tree/master/star_cfq

| Dataset Statistics | CFQ | U-CFQ | X-CFQ |
|---|---|---|---|
| Unique questions | 239,357 | 9,925,221 | 9,879,894 |
| Question patterns | 49,320 | 319,407 | 713,137 |
| Unique queries | 228,149 | 6,551,678 | 7,249,705 |
| Query patterns | 34,921 | 762,680 | 1,847,555 |
| Open questions | 108,786 | 4,658,177 | 3,879,020 |
| Closed questions | 130,571 | 5,267,044 | 6,000,814 |

Table 1: Statistics of the largest of the *-CFQ datasets, in comparison with the original CFQ. "Question pattern" here corresponds to "Question patterns (mod entities, verbs, etc.)" from Keysers et al. (2020), while "Query pattern" corresponds to "Query patterns (mod entities and properties)".

- **Expanded rule set:** The datasets span a range of rules scopes, based on grammar extensions to support additional Freebase types and properties (via new *leaf rules*) and additional syntactic constructs (via *non-leaf rules*).

All datasets in the suite include detailed instrumentation of the compositional structure of each example, in a similar format to that used in CFQ.

### 3.1 Increased Data Size

The *-CFQ datasets are generated following the algorithm described in Keysers et al. (2020) using rule sets closely related to the CFQ rules, but with sampling run at a larger scale in order to generate significantly larger datasets. As in Keysers et al. (2020), after the initial sampling phase, we apply sub-sampling and then semantic and structural filtering to increase the diversity of rule combinations while maintaining a balance of complexity levels and reducing the number of unnatural-sounding questions. The one difference from Keysers et al. (2020) is that in order to avoid an observed performance bottleneck, we omit the step of grounding the question in Freebase, which means that the generated questions contain only entity placeholders, without the guarantee that an actual set of entities can be found in Freebase that would lead to a non-empty answer to the question. In Appendix B, we show that while omitting the grounding step leads to a higher incidence of semantically implausible questions, the behavior of the baseline ML systems are highly consistent between the grounded and ungrounded datasets, which motivates our choice to use ungrounded datasets as proxies for grounded ones when exploring the behavior of ML systems at larger scales of training data.

We apply this procedure first to a nearly identical rule set as CFQ to generate the large ungrounded dataset **U-CFQ** (see Appendix D.1 for details). Datasets generated by the same procedure applied to different rule sets are described below in Section 3.3. Table 1 shows summary size statistics of two datasets from *-CFQ compared with CFQ. The size statistics for all the *-CFQ datasets can be found in Appendix C.

### 3.2 Extended Rule Set

In order to simulate coverage of a greater scope of natural language, we enrich the CFQ grammar to include up to 92%

| |
|---|
| **In** which TV program did M1 **play** |
| Did M0 **direct M2 and marry M1** |
| Who was a **crime fiction film**'s Indian writer |
| What was M1's Anglican art director's **ethnicity** |
| Was the **daughter** of the **brother** of M0 M2 |

Table 2: Examples of newly supported questions.

more *leaf rules*, which provide support for additional Freebase types and properties or add new surface forms, and up to 37% more *non-leaf rules*, which provide support for additional syntactic constructs. Examples of newly supported questions are presented in Table 2. More details on the new language features can be found in Appendix D.3.

### 3.3 Rule Set Lattice

As we are interested in exploring the effect of both the number of new rules added and their type (leaf vs. non-leaf), we prepare a suite of rule sets with varying number of leaf and non-leaf rules, which together form a *rule set lattice*. We then generate a separate large-scale dataset corresponding to each rule set in the lattice. For convenience, we will use the same name (e.g., B-CFQ, X-CFQ, etc.) to refer to both the rule set and the largest dataset generated from that rule set.

At the bottom of the lattice is a base rule set which we name **B-CFQ**, containing the rules shared by all other rule sets in the lattice. This rule set is designed to be as close as possible to U-CFQ, but with some minimal adjustments to enable separate evaluation of the addition of leaf- vs. non-leaf rules (see Appendix D for details).

**L-CFQ** is a rule set containing the rules of B-CFQ plus all additional leaf rules. **N-CFQ** consists of the rules of B-CFQ plus all additional non-leaf rules. **X-CFQ** contains the union of rules from L-CFQ and N-CFQ.

In order to test the effect of adding varying numbers of rules of a similar type, we also provide rule sets **half-L-CFQ**, **half-N-CFQ**, and **half-X-CFQ**, which have only half of the additional rules of the relevant types added.

The main characteristics of the rule sets from the lattice are shown in Table 3, with more details in Appendix D.3.

## 4 Experiments on Effect of Rule Scope

*Increased rule scope yields consistently higher error rates, which are only partially offset by increased training size.*

Figure 1 plots error rates vs. training size for a Transformer evaluated on random splits of datasets of increasing rule set scope. In each experiment, the model size and number of training steps (and hence, computational cost) are held constant, with hyperparameters as described in Appendix E.

Previous research has shown a power law relationship between data size and error rates or loss across a variety of deep learning architectures and tasks when model size and data size are increased in tandem (Kaplan et al. 2020; Brown et al. 2020; Rosenfeld et al. 2019; Hestness et al. 2017).

The results shown in Figure 1 are compatible with this previous research in that for all rule scopes, error rates initially vary in a rough power law relationship with training

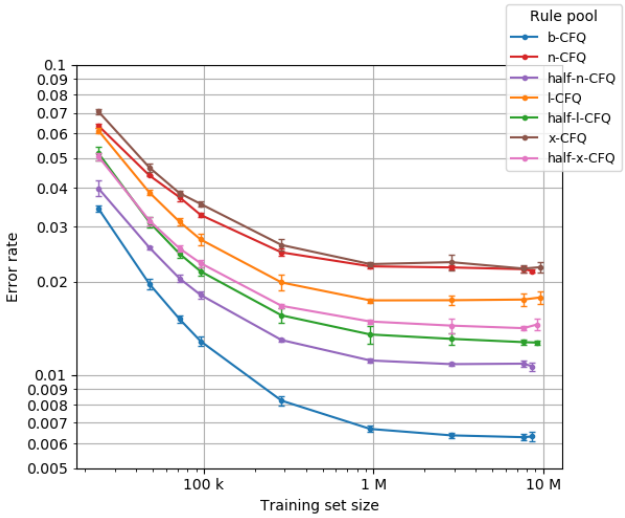


Figure 1: Effect of rule scope on scalability curve. Error rate vs train set size, all datasets from the lattice, double log scale. Every point here (and in all other graphs with random splits) averages values from 5 replicas, error bars represent error margins for confidence level 0.95.

size, while flattening out at higher training sizes. We suspect that the flattening-out is due primarily to the fixed computational cost setting, and that with sufficiently large model sizes the power law relationship would persist longer.

What is most notable in Figure 1 is that as the rule scope increases, the asymptotic value of the error rate increases consistently, such that X-CFQ plateaus at an error rate roughly 3 times that of B-CFQ. This suggests a notable scalability implication for the Transformer architecture, in that, despite the fact that every dataset in the *-CFQ suite is fully described by a small set of rules, the moderately-sized Transformer fails to fully capture the rules regardless of the number of training examples provided, and training data alone is insufficient to fully offset the increased error rates that result from even modest increases to the rule scope.

It is also notable that adding non-leaf rules impacts error rate more than adding leaf rules, as evidenced by the error rates for N-CFQ being considerably higher than L-CFQ – in fact, nearly matching the error rates of the combined rule set X-CFQ. As shown in Appendix F, this is despite the fact that the L-CFQ rule set describes a much larger space of possible questions than that described by N-CFQ. This suggests that Transformers struggle more with generalizing to new complex rule combinations than with generalizing to "primitive substitutions" (Li et al. 2019; Russin et al. 2019; Gordon et al. 2020), in which one leaf element is replaced by another within a question pattern observed in training.

## 5 Experiments on Effect of Compound Divergence

*Compound divergence dramatically affects error rates at all training sizes. Large increases in training data yield greater benefit, however, at high compound divergence than at low.*

| Rule set | Description | # Grammar rules (Non-leaf) | (Leaf) | (Total) | # All rules |
|---|---|---|---|---|---|
| CFQ | Original CFQ rule set from Keysers et al. (2020) | 54 | 157 | 211 | 443 |
| U-CFQ | Rule set for ungrounded version of CFQ | 54 | 158 | 212 | 444 |
| B-CFQ | Base CFQ | 55 | 137 | 192 | 412 |
| N-CFQ | Base CFQ + additional non-leaf rules | 70 | 139 | 209 | 455 |
| half-N-CFQ | Base CFQ + half the additional non-leaf rules | 62 | 139 | 201 | 447 |
| L-CFQ | Base CFQ + additional leaf rules | 59 | 324 | 383 | 738 |
| half-L-CFQ | Base CFQ + half the additional leaf rules | 59 | 236 | 295 | 597 |
| X-CFQ | Base CFQ + both types of additional rules | 74 | 331 | 405 | 799 |
| half-X-CFQ | Base CFQ + half of both types of additional rules | 66 | 238 | 304 | 602 |

Table 3: A list of the new CFQ-based rule sets with characteristics.

Keysers et al. (2020) observe that as the compound divergence between a train set and test set increases – that is, as the need for compositional generalization increases – the accuracy of a Transformer on the CFQ task decreases dramatically from over 98% at compound divergence 0 (a random split) to less than 20% at compound divergence 0.7 when training size is around 100k examples. In their Appendix H they also present preliminary results of the effect of training size on this performance gap – specifically, that the performance gap is even wider at smaller training sizes (e.g., around 10k examples) and shrinks as training size increases. This leaves open the question as to whether further increasing the training size beyond 100k could at some point narrow the performance gap sufficiently that the systems can be said to have effectively learned to compositionally generalize.

We investigate this question by reproducing the experiments of Keysers et al. (2020) across a wider range of training sizes, from around 10k up to nearly 900k examples. As seen in the results in Figure 2, compositional generalization remains a challenge, with even a moderate compound divergence of 0.2 yielding error rates an order of magnitude higher than those at compound divergence 0 even at the largest training sizes. Similarly to results on random splits, error rates vary in a rough power law relationship with training size in ranges of moderately large training data, while presumably plateauing at very large training sizes. However, a noticeable warm-up period can be observed, in which error rates decrease at a much slower rate, with the warm-up period persisting longer, the greater the compound divergence. Also, while in Figure 1 error rates on random splits consistently began to flatten out starting at around 100-300k training examples (with the plateau even more pronounced in Figure 2 for the compound divergence 0 split), at the higher compound divergence levels, the plateau is yet to be seen for training sizes up to 1M. This suggests that greater benefit can be reaped from very large training sizes in scenarios requiring compositional generalization than in i.i.d. settings. Further experiments at larger training sizes would be required to verify how much compound divergence affects the final asymptotic error level.

A more detailed comparison of our investigation with that of Keysers et al. (2020) is covered in Appendix G.

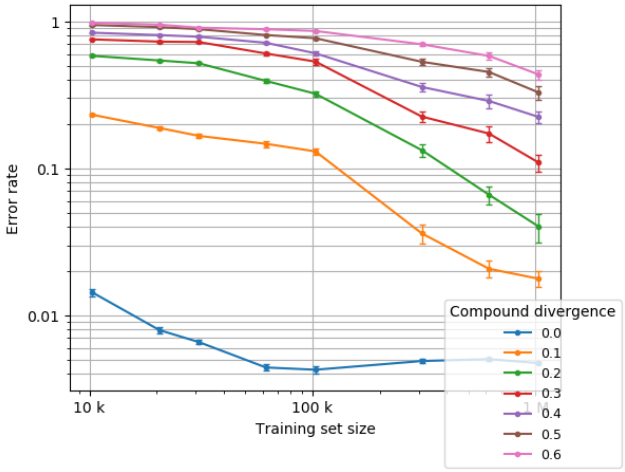


Figure 2: Effect of compound divergence on scalability curve. Error rate vs train set size for compound divergence splits for U-CFQ. Every point averages 5 replicas of up to 36 splits, generated using different random seeds; error bars represent error margins for confidence level 0.95.

## 6 Experiments on Effect of Data from Related Domain

While the experiment results from Section 4 show a clear trend of increasing error rate as rule scope increases, this trend can be seen as the combined effect of multiple simultaneously changing factors, which we may group roughly into *test set effects* and *train set effects*.

While increasing rule scope affects the test set on one hand in the obvious way of increasing its breadth, there may be more subtle effects that could lead to either an increase or decrease in inherently "easy" or "hard" examples in the test set, depending on which existing rules the newly added rules tend to compose most easily with. From the perspective of investigating the scalability of ML systems, these test set effects are of limited interest, as they largely represent artifacts of the specific data generation and sampling algorithm.

Of greater interest are the train set effects. On one hand, we expect increasing the rule scope to *dilute* the train set, in that, from the perspective of any given test example, the pro-

portion of train examples directly relevant to it (e.g., which use a large fraction of the rules or combinations of rules that appear in it) will decrease as the rule scope covered by the train set increases. At the same time, however, increasing the rule scope may increase the diversity of contexts in which each rule is seen, which may improve generalization (Hill et al. 2019). A key question governing how well ML systems can scale to larger scopes of natural language is indeed how much benefit the system can derive from these training examples that are only partially or indirectly relevant to any given subset of test examples.

To better answer this question while minimizing interference from test set effects, we shift our focus to the following experiment setup. We choose as point of reference a fixed test set $W_{TEST}$, which is randomly sampled from some limited-scope *target domain* $W$ – specifically, here we will use the B-CFQ dataset. We then suppose we have access to a limited set $W_{TRAIN}$ of $n_{in}$ training examples drawn from the identical distribution as $W_{TEST}$, plus a potentially larger supplementary set $V_{TRAIN}$ of $n_{sup}$ training examples sampled from a *related domain* $V$ that is related to $W$ through some overlap in the rule sets used to generate them. $V$ might consist, for example, of other movie-related questions that use some different syntactic constructs (i.e., different non-leaf rules) that are not present in $W$, or of questions that follow a similar syntax but include words referring to different Freebase properties and types (i.e. different leaf rules). We investigate, for various choices of supplementary domain $V$ and values of $n_{in}$ and $n_{sup}$, the degree to which blending the supplementary data set $V_{TRAIN}$ into the train set improves or harms performance on the original test set $W_{TEST}$.

In each case, we use as supplementary domain $V$ one of the datasets from the *-CFQ rule set lattice – specifically one of half-L-CFQ, L-CFQ, half-X-CFQ or X-CFQ – with examples generatable by the B-CFQ rule set filtered out. We omit experiments involving half-N-CFQ or N-CFQ, which do not lend themselves well to this experiment setup, due to the high overlap between their domains and B-CFQ. (See Appendix J for details.)

For all the experiments we use an approach similar to the one in Keysers et al. (2020):

- We use a Transformer architecture with hyperparameters described in Appendix E.
- For each split we train 5 replicas and average the results.

Details of how the train sets $W_{TRAIN}$ and $V_{TRAIN}$ are blended are described in Appendix H.

In all experiments we used a fixed test set of $95{,}742$ examples sampled randomly from B-CFQ.

### 6.1 Equal Weighting of Examples from Target and Related Domains

*Transformers are sensitive to the training distribution, not just the information contained in the training data. When highly data-starved, adding training examples from a related domain can improve performance. As training size increases, however, the skew in train vs. test distribution resulting from expansion of the train set quickly outweighs any benefits from the additional training examples.*

In this experiment, we weight examples from $W_{TRAIN}$ and $V_{TRAIN}$ equally, so that as the number of examples in $V_{TRAIN}$ increases, we can observe the same dynamics we would see when scaling to increasing scopes of language, where expanding the scope of the train set simultaneously increases the amount of total information in the train set while also diluting the fraction of the train set that is directly relevant to any given slice of test examples.

Figure 3(a) summarizes the results when $n_{in}$ is fixed at 10k examples and $n_{sup}$ varies between 10k and 8M.

As in Figure 1, error rate varies with training size in a clear power law relation for training sizes up to around 100k-300k (depending on the dataset), after which performance plateaus. However, the more distant the related domain is from the target domain in terms of number of leaf or non-leaf rules added, the more slowly the error rates drop, and the higher the asymptotic value at which they plateau. This reinforces that the increased error rates observed in Section 4 for increased rule scope are indeed driven largely by the changing composition of the train set, rather than purely by test set effects.

For easier comparison with Figure 1, we mark on each curve the point at which the relative size of $n_{sup}$ vs. $n_{in}$ matches the ratio that would be expected based on the relative sizes of the domains of $V$ vs. $W$, if the train set were sampled from $V$ as a whole, while the test set were simply fixed to observe the performance on the subset of $V$ generated by the narrower rule set $W$. It can be seen that the error rates at the marked points in Figure 3 are slightly higher than at similar training sizes in Figure 1, suggesting that a small degree of test set effect is also at play.

Figure 3(b) summarizes the results when $n_{in}$ is fixed at the larger value of 100k examples, while $n_{sup}$ varies as before between 10k and 8M. Here, unlike in the data-starved scenario of Figure 3(a), the benefit from the additional training data is more limited, with error rates actually increasing when $n_{sup}$ is significantly larger than $n_{in}$. This suggests that as training size increases, the skew in train vs. test distribution resulting from expansion of the train set quickly outweighs any benefits from the additional training examples. Notably, by the time that $n_{sup}$ approaches 100x the value of $n_{in}$, error rates have already increased to levels comparable to those at which they are seen to plateau in Figure 3(a) for the same choice of $V$, despite the order of magnitude difference in the amount of in-domain training data.

Results for other choices of $n_{in}$ ranging from 10k up through 500k examples are presented in Appendix I.

### 6.2 Over-weighting of Examples from Target Domain

*If sample weighting is controlled, then additional data from a related domain can be made to yield a strictly positive effect. However, the performance benefits achievable through this method are limited.*

If we had the luxury of being able to train a separate model for each domain, we could consider mitigating the

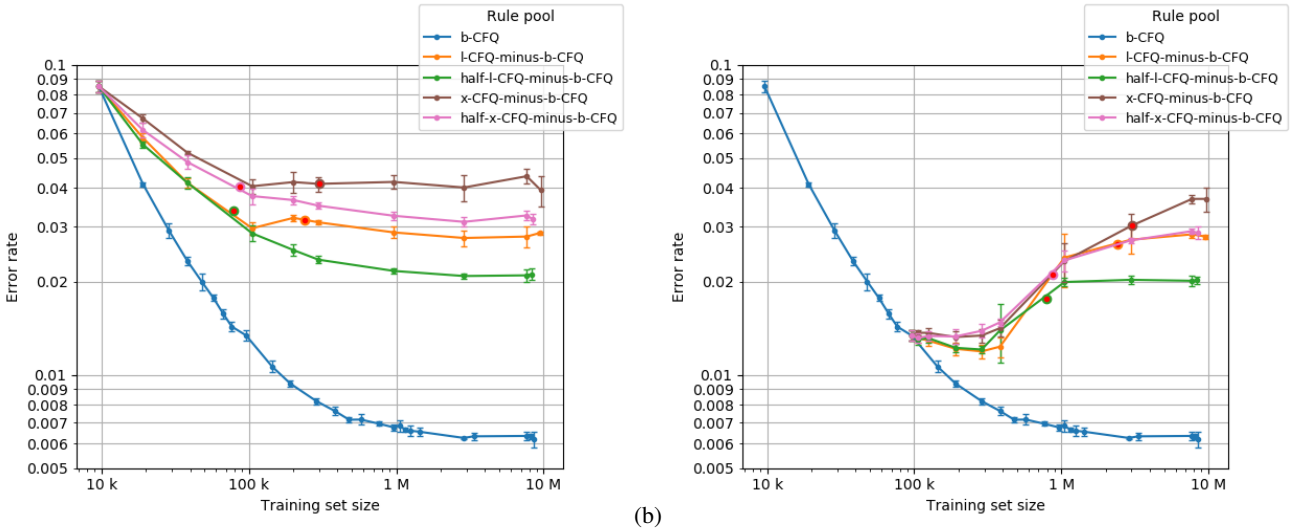


Figure 3: Error rate vs train set size, log-log scale, equal weighting of examples from target and related domains: 10k examples (a) and 100k examples (b), B-CFQ test set, B-CFQ + others train set.

harm caused by the skew in train-test distribution seen in Section 6.1 by giving special treatment to the in-domain train examples. Here we evaluate the effect of one form of such special treatment by increasing the sampling weight of examples from $W_{TRAIN}$ relative to those from $V_{TRAIN}$ to ensure that at least half of all training observations correspond to in-domain examples. The target ratio of one-half matches the optimal ratio observed by Wang et al. (2017) for a similar instance weighting technique.

Specifically, if, for example, $n_{in}$ is 10k and $n_{sup}$ is 100k, we would duplicate each example of $W_{TRAIN}$ 10 times (prior to shuffling), so that when randomly sampling from the adjusted train set, exactly half of the samples are from $W_{TRAIN}$ and half are from $V_{TRAIN}$. If $n_{sup}$ is less than $n_{in}$, then no adjustment is made.

Figures 4(a) and (b) again show the results when $n_{in}$ is fixed at 10k and 100k examples, respectively.

As the figures show, if sample weighting is controlled, then additional data from a related domain can be made in most cases to yield a strictly positive effect.

However, again, the more distant the related domain is from the target domain, the more slowly the error rates drop – and most significantly, the higher the asymptotic value at which they plateau. While we expect that error rates could be further improved by optimizing the exact sampling ratio between $W_{TRAIN}$ and $V_{TRAIN}$, the results so far suggest that there are limits to the performance benefits achievable through this method.

## 7 Related Work

The relationship between training size and error rates or generalization loss has been studied in a variety of settings. Early research observed that accuracy varies roughly with the log of the training size, both in shallow ML approaches on a natural language disambiguation task (Banko and Brill 2001) and in convolutional neural networks on an object detection task (Sun et al. 2017). Others, noting the connection with learning curves observed during training on a large train set, have tried fitting a power law learning curve to the relationship between training size and error rate (Figueroa et al. 2012). More recent research has observed consistent power law relationships between training size and cross-entropy error across a range of deep learning architectures, including LSTMs on language modeling and machine translation tasks (Hestness et al. 2017) and LSTMs or Transformers on language modeling tasks (Rosenfeld et al. 2019; Kaplan et al. 2020; Brown et al. 2020), provided that training size and model size are increased in tandem. In particular Kaplan et al. (2020) observe that a highly consistent power law relationship between cross-entropy loss and either of training size or model size persists across many orders of magnitude when not bottlenecked by the other of the two. Our approach differs from this previous research by specifically investigating the effect of the compositional structure of a task on the training size to error rate relationship.

The degree to which additional training data from a related domain can help or hurt performance has been investigated also in the context of multi-domain learning (Yang and Hospedales 2014; Herzig and Berant 2017; Britz, Le, and Pryzant 2017; Tars and Fishel 2018; Mghabbar and Ratnamogan 2020; Wang et al. 2020) and domain adaptation (Wang et al. 2017; Chu and Wang 2018; Zhang et al. 2019; Wilson and Cook 2020).

Our experiments in Section 6.1 resemble the scenario of multi-domain learning (MDL) in that we aim to train a single model that can apply to multiple domains, not just the single domain from which our test set is drawn. Multi-domain learning seeks, however, to improve performance on individual domains by explicitly distinguishing the domain of each example at train and test time, which is not our focus here. We expect that explicitly distinguishing domains would be less beneficial in the *-CFQ tasks than in scenarios where

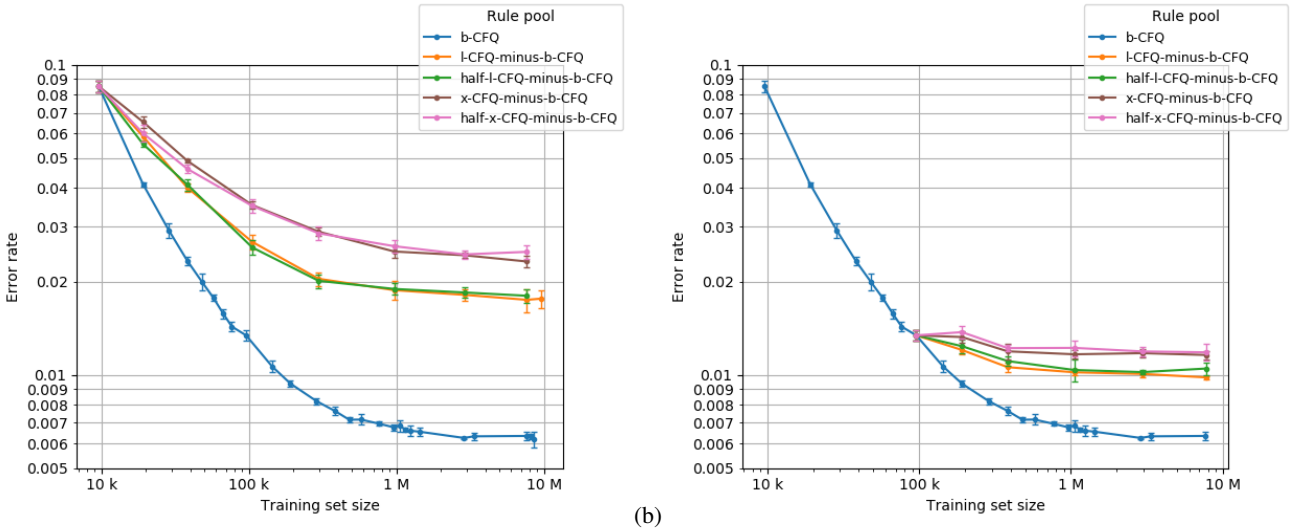


Figure 4: Error rate vs train set size, log-log scale, fixed ratio: 10k B-CFQ examples (a) and 100k B-CFQ examples (b), B-CFQ test set, B-CFQ + others train set.

the expected output for a given input can differ depending on the domain (something that does not occur in *-CFQ), but we would welcome investigation into the effectiveness of MDL techniques on *-CFQ as future work.

Our experiments in Section 6.2 can be considered a form of domain adaptation (DA) in that a model targeting a single domain is trained using a combination of in-domain and out-of-domain data. Our approach corresponds most closely to the *instance weighting* approach, particularly the *batch weighting* variant, which Wang et al. (2017) found to yield superior performance over other instance weighting techniques. Another popular DA technique which we have not evaluated in our experiments is that of pre-training on a broader data set followed by fine-tuning on in-domain data (Luong and Manning 2015), which may be augmented with selections from the out-of-domain data ranked by similarity to the in-domain distribution (Zhang et al. 2019).

In particular, much attention has been paid recently to the significant improvements in performance achievable on a variety of downstream natural language tasks through pre-training of large scale language models (Devlin et al. 2018; Raffel et al. 2019; Yang et al. 2019; Brown et al. 2020), including in scenarios of domain adaptation and multi-domain learning (Talmor and Berant 2019; Gururangan et al. 2020). Talmor and Berant (2019) observe that the performance of BERT-large (Devlin et al. 2018) can be improved on several reading comprehension (RC) benchmarks by additionally pre-training it on a selection of 75k supervised examples from each of 5 different large RC datasets, prior to finally fine-tuning it on the specific target dataset. In cases where the target dataset is large, however, they find that the additional pre-training from the other RC datasets improves performance only about half the time. Gururangan et al. (2020) show that applying additional pre-training on unlabeled examples from the target domain improves performance on a number of classification tasks compared to use of RoBERTa (Liu et al. 2019) alone.

Our approach differs from this previous research again by using knowledge of the compositional structure of the task to characterize the relationship between the target domain and related domains, and illustrating the effect of these factors on the slope and limit value of the performance curves.

## 8 Conclusion and Outlook

In this paper we present *-CFQ, a suite of large-scale datasets designed for principled investigation into the scalability of ML systems on a compositional NLU task. To the best of our knowledge, *-CFQ is uniquely suited for this investigation, because it is the first dataset to achieve this scale while providing full details on the compositional structure of each example. We use this dataset to perform experiments illustrating that scalability is indeed a concern even at a scope that is only a tiny fraction of full natural language.

The experiments presented in this paper, however, only scratch the surface of the types of controlled investigations of ML scaling behavior possible using *-CFQ. We hope that this dataset suite will aid the deep learning and NLU communities in the development of more robust and scalable solutions to language understanding.

In particular, we hope to re-use *-CFQ to evaluate the degree to which language model pre-training and increase of model size affect the scalability curves in the compositional setting. We are interested in exploring the combined effects of compound divergence and rule scope. Further, we are interested to estimate the necessary data sizes and computational power that would be required by current ML approaches to achieve robust levels of language understanding across arbitrary domains and syntax, based on the above results together with an estimate of the effective number of language features in full natural language compared to X-CFQ.

## Acknowledgements

We thank Daniel Furrer for code reviews, helpful input in weekly syncs, improving the efficiency of the compound divergence splitting algorithm, and help in running or debugging ML experiments; Xiao Wang for contributing to the planning of grammar extensions; Olivier Bousquet for providing guidance and insights; and Daniel Keysers and the anonymous reviewers for helpful feedback on the paper.

## Ethics Statement

The purpose of this paper is well aligned with sustainability and efficient use of resources. More scalable models need fewer training examples and thus fewer resources to reach the same accuracy than less scalable ones.

Since the *-CFQ datasets are generated and ungrounded (except for O-CFQ), they do not contain any personal data. O-CFQ does not contain private data as it is grounded on Freebase which is publicly available.

In general, when preparing datasets for use in training language models, care should be taken to avoid inclusion of potentially harmful biases which may be propagated to the language models. Due to their systematic, rule-based generation method, the risk of unintended bias in *-CFQ, as in CFQ, is low compared to datasets mined from large corpora of text in the wild. The main source of potential bias would be in the distribution of entities in the Freebase dataset itself, which may be propagated to the resulting generated questions, particularly during the grounding step. The risk of such bias is reduced in *-CFQ through the use of ungrounded questions in most of its datasets. In the choice of adjectives modeled in the dataset, some bias (or loss of comprehensiveness) is also introduced due to our choice of only those adjectives whose corresponding entities are relatively frequently used in Freebase.

## References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*. URL http://arxiv.org/abs/1409.0473.

Banko, M.; and Brill, E. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, 26–33.

Battaglia, P. W.; Hamrick, J. B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. 2018. Relational inductive biases, deep learning, and graph networks. *CoRR* abs/1806.01261. URL https://arxiv.org/pdf/1806.01261.pdf.

Britz, D.; Le, Q.; and Pryzant, R. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, 118–126.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* .

Cho, J.; Lee, K.; Shin, E.; Choy, G.; and Do, S. 2015. How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? *arXiv preprint arXiv:1511.06348* .

Chomsky, N. 1956. Three models for the description of language. *IRE Transactions on information theory* 2(3): 113–124.

Chu, C.; and Wang, R. 2018. A survey of domain adaptation for neural machine translation. *arXiv preprint arXiv:1806.00258* .

Dehghani, M.; Gouws, S.; Vinyals, O.; Uszkoreit, J.; and Kaiser, Ł. 2018. Universal transformers. *CoRR* abs/1807.03819. URL https://arxiv.org/pdf/1807.03819.pdf.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Figueroa, R. L.; Zeng-Treitler, Q.; Kandula, S.; and Ngo, L. H. 2012. Predicting sample size required for classification performance. *BMC medical informatics and decision making* 12(1): 8.

Fodor, J. A.; Pylyshyn, Z. W.; et al. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition* 28(1-2): 3–71.

Freivalds, K.; Ozoliņš, E.; and Šostaks, A. 2019. Neural Shuffle-Exchange Networks-Sequence Processing in O (n log n) Time. In *Advances in Neural Information Processing Systems*, 6630–6641.

Furrer, D.; van Zee, M.; Scales, N.; and Schärli, N. 2020. Compositional Generalization in Semantic Parsing: Pre-training vs. Specialized Architectures. *arXiv preprint arXiv:2007.08970* .

Gordon, J.; Lopez-Paz, D.; Baroni, M.; and Bouchacourt, D. 2020. Permutation Equivariant Models for Compositional Generalization in Language. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=SylVNerFvr.

Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; and Smith, N. A. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. *arXiv preprint arXiv:2004.10964* .

Herzig, J.; and Berant, J. 2017. Neural semantic parsing over multiple knowledge-bases. *arXiv preprint arXiv:1702.01569* .

Hestness, J.; Narang, S.; Ardalani, N.; Diamos, G.; Jun, H.; Kianinejad, H.; Patwary, M.; Ali, M.; Yang, Y.; and Zhou, Y. 2017. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409* .

Hill, F.; Lampinen, A.; Schneider, R.; Clark, S.; Botvinick, M.; McClelland, J. L.; and Santoro, A. 2019. Emergent systematic generalization in a situated agent. *arXiv preprint arXiv:1910.00571* .

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Hupkes, D.; Dankers, V.; Mul, M.; and Bruni, E. 2019. The compositionality of neural networks: integrating symbolism and connectionism. *arXiv preprint arXiv:1908.08351* .

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* .

Keysers, D.; Schärli, N.; Scales, N.; Buisman, H.; Furrer, D.; Kashubin, S.; Momchev, N.; Sinopalnikov, D.; Stafiniak, L.; Tihon, T.; Tsarkov, D.; Wang, X.; van Zee, M.; and Bousquet, O. 2020. Measuring Compositional Generalization: A Comprehensive Method on Realistic Data. In *ICLR*. URL https://arxiv.org/abs/1912.09713.pdf.

Lake, B. M.; and Baroni, M. 2018. Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks. In *ICML*. URL https://arxiv.org/pdf/1711.00350.pdf.

Li, Y.; Zhao, L.; Wang, J.; and Hestness, J. 2019. Compositional Generalization for Primitive Substitutions. *arXiv preprint arXiv:1910.02612* .

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* .

Luong, M.-T.; and Manning, C. D. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, 76–79.

Mghabbar, I.; and Ratnamogan, P. 2020. Building a Multi-domain Neural Machine Translation Model using Knowledge Distillation. *arXiv preprint arXiv:2004.07324* .

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* .

Rosenfeld, J. S.; Rosenfeld, A.; Belinkov, Y.; and Shavit, N. 2019. A constructive prediction of the generalization error across scales. *arXiv preprint arXiv:1909.12673* .

Russin, J.; Jo, J.; O'Reilly, R. C.; and Bengio, Y. 2019. Compositional generalization in a deep seq2seq model by separating syntax and semantics. *arXiv preprint arXiv:1904.09708* .

Sun, C.; Shrivastava, A.; Singh, S.; and Gupta, A. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, 843–852.

Talmor, A.; and Berant, J. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. *arXiv preprint arXiv:1905.13453* .

Tars, S.; and Fishel, M. 2018. Multi-domain neural machine translation. *arXiv preprint arXiv:1805.02282* .

Vaswani, A.; Bengio, S.; Brevdo, E.; Chollet, F.; Gomez, A. N.; Gouws, S.; Jones, L.; Kaiser, Ł.; Kalchbrenner, N.; Parmar, N.; et al. 2018. Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416* .

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*. URL https://arxiv.org/pdf/1706.03762.pdf.

Wang, R.; Utiyama, M.; Liu, L.; Chen, K.; and Sumita, E. 2017. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1482–1488.

Wang, W.; Tian, Y.; Ngiam, J.; Yang, Y.; Caswell, I.; and Parekh, Z. 2020. Learning a Multi-Domain Curriculum for Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7711–7723.

Wilson, G.; and Cook, D. J. 2020. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)* 11(5): 1–46.

Yang, Y.; and Hospedales, T. M. 2014. A unified perspective on multi-domain and multi-task learning. *arXiv preprint arXiv:1412.7489* .

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, 5753–5763.

Zhang, X.; Shapiro, P.; Kumar, G.; McNamee, P.; Carpuat, M.; and Duh, K. 2019. Curriculum learning for domain adaptation in neural machine translation. *arXiv preprint arXiv:1905.05816* .