Good-Enough Compositional Data Augmentation

Jacob Andreas
MIT CSAIL
jda@mit.edu

Abstract

We propose a simple data augmentation protocol aimed at providing a compositional inductive bias in conditional and unconditional sequence models. Under this protocol, synthetic training examples are constructed by taking real training examples and replacing (possibly discontinuous) fragments with other fragments that appear in at least one similar environment. The protocol is model-agnostic and useful for a variety of tasks. Applied to neural sequence-to-sequence models, it reduces error rate by as much as 87% on diagnostic tasks from the SCAN dataset and 16% on a semantic parsing task. Applied to n-gram language models, it reduces perplexity by roughly 1% on small corpora in several languages.

1 Introduction

This paper proposes a rule-based data augmentation protocol for sequence modeling. Our approach aims to supply a simple and model-agnostic bias toward compositional reuse of previously observed sequence fragments in novel environments. Consider a language modeling task in which we wish to estimate a probability distribution over a family of sentences with the following finite sample as training data:

(1) a. *The cat sang.*
b. *The wug sang.*
c. *The cat daxed.*

In language processing problems, we often want models to generalize beyond this dataset and infer that (2a) is also probable but (2b) is not:

(2) a. *The wug daxed.*
b. * *The sang daxed.*

This generalization amounts to an inference about syntactic categories (Clark, 2000). Because *cat* and *wug* are interchangeable in (1a) and (1b), they are also likely interchangeable elsewhere; *cat* and *sang* are not similarly interchangeable. Human learners make judgments like (2) about novel lexical items (Berko, 1958) and fragments of novel languages (Lake et al., 2019). But we do not expect such judgments from unstructured generative models trained to maximize the likelihood of the training data in (1).

A large body of work in natural language processing provides generalization to data like (2a) by adding structure to the learned predictor (Chelba and Jelinek, 1998; Chiang, 2005; Dyer et al., 2016). On real-world datasets, however, such models are typically worse than "black-box" function approximators like neural networks, even for black-box models that fail to place probability mass on either example in (2) given small training sets like (1) (Melis et al., 2018). To the extent that we believe (2a) to capture an important inductive bias, we would like to find a way of softly encouraging it without tampering with the structure of predictors that work well at scale. In this paper, we introduce a procedure for generating synthetic training examples by recombining real ones, such that (2a) is assigned non-negligible probability because it *already appears in the training dataset*.

The basic operation underlying our proposal (which we call GECA, for "good-enough compositional augmentation") is depicted in Figure 1: if two (possibly discontinuous) fragments of training examples appear in *some* common environment, then *any* additional environment where the first fragment appears is also a valid environment for the second.

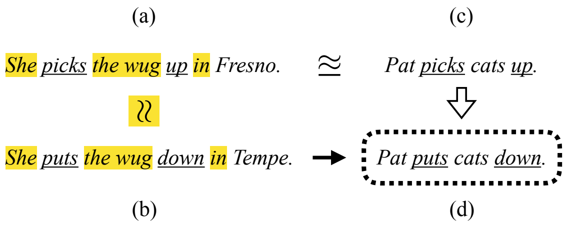
(a)
(c)
She picks the wug up in Fresno.
≈
Pat picks cats up.
≈
She puts the wug down in Tempe.
Pat puts cats down.
(b)
(d)

Figure 1: Visualization of the proposed approach: two discontinuous sentence fragments (a–b, underlined) which appear in similar environments (a–b, highlighted) are identified. Additional sentences in which the first fragment appears (c) are used to synthesize new examples (d) by substituting in the second fragment.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7556–7566
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

GECA is crude: as a linguistic principle, it is both limited and imprecise. As discussed in Sections 3 and 4, it captures a narrow slice of the many phenomena studied under the heading of "compositionality", while also making a number of incorrect predictions about real language data. Nevertheless, GECA appears to be quite effective across a range of learning problems. In semantic parsing, it gives improvements comparable to the task-specific data augmentation approach of Jia and Liang (2016) on logical expressions, better performance than that approach on a different split of the data designed to test generalization more rigorously, and corresponding improvements on a version of the dataset with a different meaning representation language. Outside of semantic parsing, it solves two representative problems from the SCAN dataset of Lake and Baroni (2018) that are synthetic but precise in the notion of compositionality they test. Finally, it helps with some (unconditional) low-resource language modeling problems in a typologically diverse set of six languages.

2 Background

Recent years have seen tremendous success at natural language transduction and generation tasks using complex function approximators, especially recurrent (Sutskever et al., 2014) and attentional (Vaswani et al., 2017) neural models. With enough training data, these models are often more accurate than than approaches built on traditional tools like regular transducers and context-free grammars (Knight and Graehl, 2005), which are brittle and difficult to efficiently infer from large datasets.

However, models equipped with an explicit symbolic generative process have at least one significant advantage over the aforementioned black-box approaches: given a grammar, it is straightforward to precisely characterize how that grammar will extrapolate beyond the examples in a given training set to out-of-distribution data. Indeed, it is often possible for researchers to design the form that this extrapolation will take: smoothed n-gram language models ensure that no memorization is possible beyond a certain length (Ney et al., 1994); CCG-based semantic parsers can make immediate use of entity lexicons without having ever seen the lexicon entries used in real sentences (Zettlemoyer and Collins, 2005).

It is not the case that black-box neural models are fundamentally incapable of this kind of predictable generalization—the success of these models at capturing long-range structure in text (Radford et al., 2019) and controlled algorithmic data (Graves et al., 2014) indicate that some representation of hierarchical structure can be learned given enough data. But the precise point at which this transition occurs is not well characterized, and it is evidently beyond the scale available in many real-world problems.

How can we improve the behavior of high-quality black-box models in these settings? There are many sophisticated tools available for improving the function approximators or loss functions themselves—structured regularization of parameters (Oh et al., 2017), posterior regularization (Ganchev et al., 2010; Hu et al., 2018), explicit stacks (Grefenstette et al., 2015) and composition operators (Bowman et al., 2016; Russin et al., 2019). These existing proposals tend to be task- and architecture-specific. But to the extent that the generalization problem can be addressed by increasing the scale of the training data, it is natural to ask whether we can address the problem by increasing this scale *artificially*—in other words, via data augmentation.

Data augmentation techniques, which generate auxiliary training data by performing structured transformation or combination of training examples, are widely used in computer vision (Krizhevsky et al., 2012; Zhang et al., 2017; Summers and Dinneen, 2019). Within NLP, several data augmentation approaches have been proposed for text classification (e.g. Ratner et al., 2017; Wei and Zhou, 2019); these approaches give improvements even when combined with large-scale pretraining (Hu et al., 2019). Jia and Liang (2016) study data augmentation and compositionality in specific setting of learning language-to-logical-form mappings, beginning from the principle that data is compositional if it is generated by an explicit grammar that relates strings to logical forms. This view of compositionality as determined by synchronicity between form and meaning is essentially Montagovian and well-suited to problems in formal semantics (Montague, 1973); however, it requires access to structured meaning representations with explicit types and bracketings, which are not available in most NLP applications.

Here we aim at a notion of compositionality that is simpler and more general: a bias toward identifying recurring fragments seen at training time, and re-using them in environments distinct from those in which they were first observed. This view makes no assumptions about the availability of brackets and types, and is synchronous only to the extent that the notion of a fragment is permitted to include content from both the source and target sides. We will find that it is nearly as effective as existing approaches in the specific settings for which they were designed, but also effective on a variety of problems where they cannot be applied.

3 Approach

Consider again the example in Figure 1. Our data augmentation protocol aims to discover substitutable sentence **fragments** (underlined), with the fact that a pair of fragments appear in some common sub-sentential **environment** (highlighted) taken as evidence that the fragments belong to a common category. To generate a new examples for the model, an occurrence of one fragment is removed from a sentence to produce a sentence **template**, which is then populated with the other fragment.

Why should we expect this procedure to produce well-formed training examples? The existence of syntactic categories, and the expressibility of well-formedness rules in terms of these abstract categories, is one of the foundational principles of generative approaches to syntax (Chomsky, 1965). The observation that context provides a strong signal about a sentence fragment's category is in turn the foundation of distributional techniques for the study of language (Firth, 1957). Combining the two gives the outlines of the above procedure.

This combination has a productive history in natural language processing: when fragments are single words, it yields class-based language models (Brown et al., 1992); when fragments are contiguous spans it yields unsupervised parsers (Clark, 2000; Klein and Manning, 2002). The present data augmentation scenario is distinguished mainly by the fact that we are *unconcerned* with producing a complete generative model of data, or with recovering the latent structure implied by the presence of nested syntactic categories. We can still synthesize high-precision examples of well-formed sequences by identifying individual substitutions that are likely to be correct without understanding how they fit into the grammar as a whole.

Indeed, if we are not concerned with recovering linguistically plausible analyses, we need not limit ourselves to words or contiguous sentence fragments. We can take

(3) a. *She picks the wug up.*
b. *She puts the wug down.*

as evidence that we can use *picks... up* wherever we can use *puts... down*. Indeed, given a *translation* dataset:

(4) a. *I sing.* ▷ *Canto.*
b. *I sing marvelously.* ▷
Canto maravillosamente.
c. *I dax marvelously.* ▷
Dajo maravillosamente.

we can apply the same principle to synthesize *I dax.* ▷ *Dajo.* based on the common environment *... marvelously* ▷ *... maravillosamente*. From the perspective of a generalized substitution principle, the alignment problem in machine translation is the same as the class induction problem in language modeling, but with sequences featuring large numbers of gappy fragments and a boundary symbol ▷.

The only remaining question is what makes two environments similar enough to infer the existence of a common category. There is, again, a large literature on this question (including the aforementioned work in language modeling, unsupervised parsing, and alignment), but in the current work we will make use of a very simple criterion: fragments are interchangeable if they occur in at least one **lexical environment** that is exactly the same.

Given a **window size** k and sequence of n tokens $w = w_1 w_2 \cdots w_n$, define a **fragment** as a set of non-overlapping spans of w, a **template** as a version of w with a fragment removed, and an **environment** as a template restricted to a k-word window around each removed fragment. Formally, (letting $[i, j]$ denote $\{i, i+1, \ldots, j\}$) we have:

$$\begin{aligned}\texttt{fragments}(w) = \{\{w_{a_1..b_1}, w_{a_2..b_2}, \ldots\} :& \\ 1 \leq a_i < b_i \leq n, \text{all } [a_i, b_i] \text{ disjoint}\}&\end{aligned} \tag{1}$$

$$\texttt{tpl}(w, f) = (w_j : \forall w_{a_i..b_i} \in f.\ j \notin [a_i, b_i]) \tag{2}$$

$$\begin{aligned}\texttt{env}(w, f) = \{w_j :& \\ & w_j \in \texttt{tpl}(w, f) \text{ and} \\ & \exists w_{a_i..b_i} \in f.\ j \in [a_i - k, b_i + k]\}\end{aligned} \tag{3}$$

In Figure 1(a), the underlined <u>picks</u>... <u>up</u> is one possible fragment that could be extracted from the sentence. The corresponding template is *She... the wug ... in Fresno*, and with $k = 1$ the environment is *She. . . the wug . . . in.* As shown in Figure 1(d), any fragment may be substituted into any template with the same number of holes. Denote this substitution operation by t/f. The data augmentation operation that defines GECA is formally stated as follows:

> If the training data contains sequences $w = t_1/f_1$, $x = t'_1/f_1$ and $y = t_2/f_2$, with $env(w, t_1) = env(y, t_2)$ and $t'_1 \neq t_1$, then synthesize a new training example $z = t'_1/f_2$.

If a fragment occurs multiple times within a given example, all instances are replaced (see Figure 3).

Linguistic notes Despite the fact that the above operation is motivated by insights from generative syntax and distributional semantics, it should be emphasized that it is, as a statement of a general linguistic principle, obviously wrong. Counterexamples abound: in English, stress-derived nouns (e.g. *récord* from *recórd*) will be taken as evidence that many nouns and verbs are interchangeable; in Mandarin Chinese, *kěshì* and *dànshì* both mean "but", but *kěshì* alone can be used in particular constructions to mean "very".

What ultimately matters is the relative frequency of such errors: if their contribution to an inaccurate model is less than the inaccuracy caused by the original shortage of training data, the GECA still helps. In conditional problems, like the machine translation example above, such errors may be totally harmless: if we synthesize a new (x, y) pair with x outside the support of the real training data, they may not influence the model's predictions on the true support beyond providing useful general inductive bias.

Implementation Naïve implementation of the boxed operation takes $O(t^3 f^3)$ time (where t is the number of distinct templates in the dataset and f the number of distinct fragments). This can be improved to $O(ft^2 e)$ (where e is the number of templates that map to the same environment) by building appropriate data structures (Algorithm 1).

Space requirements might still be considerable (comparable to those used by n-gram language models), and strategies from the language modeling literature can be used to reduce memory usage (Heafield, 2011). This algorithm is agnostic with respect to the choice of fragmentation and environment functions; task-specific choices are described in more detail for each experiment below.

4 Diagnostic experiments

We begin with a set of experiments on synthetic data designed to precisely test whether GECA provides the kind of generalization it was designed for. Here we use the SCAN dataset (Lake and Baroni, 2018), which consists of simple English commands paired with sequences of discrete actions (Figure 2). We focus specifically on the *add primitive (jump)* and *add template (around right)* conditions, which test whether the agent can be exposed to individual commands or modifiers (e.g. *jump* ▷ JUMP) in isolation at training time, and incorporate them into more complex commands like the earlier example at test time.

We extract fragments with one gap and a maximum length of 4 tokens. The environment is taken to be the complete template. Generated examples are appended to the original dataset. As an exam-

Algorithm 1 Sample GECA implementation.

```
f2t = dict(default=set()) # fragment -> template
t2f = dict(default=set()) # template -> fragment
e2t = dict(default=set()) # env -> template
for seq in dataset:
  for f in fragments(seq): # Eq. 1
    template = tpl(seq, fragment) # Eq. 2
    add(f2t[fragment], template)
    add(t2f[template], fragment)
    add(e2t[env(seq, fragment)], template) # Eq. 3

t2t = dict(default=set())
for fragment in keys(f2t)):
  for template in f2t[fragment]:
    for template2 in f2t[fragment]:
      for new_template in e2t[env(template2)]
        add(t2t[template1], new_template)

for template1, template2 in t2t:
  for fragment in t2f[template1]
    if fragment not in t2f[template2]:
      yield template2 / fragment
```

walk
WALK

walk left twice
LTURN WALK LTURN WALK

jump
JUMP

jump around left
LTURN JUMP LTURN JUMP LTURN JUMP LTURN JUMP

walk right
RTURN WALK

Figure 2: Example SCAN data. Each example consists of a synthetic natural language command (left) paired with a discrete action sequence (right).

	jump / SCAN	jump / NACS	right / SCAN	right / NACS
seq2seq	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
+ GECA	**0.87** ± 0.02	**0.67** ± 0.01	**0.82** ± 0.04	**0.82** ± 0.03

Table 1: Sequence match accuracies on SCAN datasets, in which the learner must generalize to new compositional uses of a single lexical item ("jump") or multi-word modifier ("around right") when mapping instructions to action sequences (SCAN) or vice-versa (NACS, Bastings et al., 2018). While the sequence-to-sequence model is unable to make any correct generalizations at all, applying GECA enables it to succeed most of the time. Scores are averaged over 10 random seeds; the standard deviation across seeds is shown. All improvements are significant (paired binomial test, $p \ll 0.001$).

add primitive (jump)

walk thrice after walk right
RTURN WALK WALK WALK WALK

jump opposite left thrice after turn opposite right
RTURN RTURN LTURN LTURN JUMP LTURN LTURN JUMP LTURN LTURN JUMP

add template (around right)

look right twice and turn opposite right twice
RTURN LOOK RTURN LOOK RTURN RTURN RTURN RTURN

run around right and walk opposite right twice
RTURN RUN RTURN RUN RTURN RUN RTURN RUN RTURN RTURN WALK RTURN RTURN WALK

Figure 3: Examples synthesized for the SCAN tasks. Underlined words belong to the filled-in fragment; the remaining text is the template. GECA synthesizes some examples that exactly capture the desired generalization, and some examples that are unrelated.

ple of the effect of this augmentation procedure, the original *jump* split has 12620 training examples; GECA generates an additional 395 using 395 distinct templates and 6 distinct fragments.

With the original and augmented datasets, we train a one-layer LSTM encoder–decoder model with an embedding size of 64, a hidden size of 512, a bidirectional encoder and an attentional decoder (Hochreiter and Schmidhuber, 1997; Bahdanau et al., 2015). The model is trained using ADAM (Kingma and Ba, 2014) with a step size of 0.001 and a dropout rate of 0.5.

Results are shown in Table 1. In line with the original experiments of Lake and Baroni, the baseline sequence-to-sequence model completely fails to generalize to the test set. Applying GECA allows the learned model to successfully make most tested generalizations across single and multi-word entries, and in both instruction-to-action and action-to-instruction directions.

Analysis: examples Some synthesized examples are shown in Figure 3. Success at the *add primitive* condition stems from the constraint that the single example usage of the primitive must still be a valid (command, action) pair, and all verbs are valid commands in isolation. Only three examples—*run* ▷ RUN, *walk* ▷ WALK and *look* ▷ LOOK—provide the evidence that GECA uses to synthesize to new usages of *jump*; if these were removed, the sequence-to-sequence model's training accuracy would be unchanged but GECA would fail to synthesize any new examples involving *jump*, and test accuracy would fall to zero. For the *add template* condition, GECA correctly replaces all occurrences of LTURN with RTURN to produce new examples of the *around right* template; this example highlights the usefulness of GECA's ability to discover discontinuous and non-context-free substitutions.

Analysis: dataset statistics To further understand the behavior of GECA, we conduct a final set of analyses quantifying the overlap between the synthesized data and the held-out data. We first measure **full example overlap**, the fraction of test examples that appear in the augmented training set. (By design, no overlap exists between the test set and the original training set.) After applying GECA, 5% of test examples for the *add primitive* condition and 1% of examples for the *add template* condition are automatically synthesized. Next we measure **token co-occurrence overlap**: we compute the set of (input or output) tokens that occur together in any test example, and then measure the fraction of these pairs that also occur together in some training example. For the *add primitive* condition, GECA increases token co-occurrence overlap from 83% to 96%; for the *add template* condition it is 100% even prior to augmentation.

It is important to note that GECA, which sees only the training set, is unaware that some subset of the data is singled out for generalization testing at evaluation time. The data augmentation protocol generates a large number of spurious training examples unrelated to the desired generalization (e.g. the first example in Figure 3); however, it also generates enough new usages of the target concept that the learner generalizes successfully.

5 Semantic parsing

Next we turn to the problem of *semantic parsing*, which has also been a popular subject of study for questions about compositionality, generalization, and data augmentation. For the reasons discussed in Section 3, we expect qualitatively different behavior from this approach on real language data without the controlled vocabulary of SCAN.

We study four versions of the GeoQuery dataset (Zelle, 1995), which consists of 880 English questions about United States geography, paired with meaning representations in the form of either logical expressions or SQL queries. The standard train–test split for this dataset ensures that no natural language *question* is repeated between the train and test sets. As Finegan-Dollak et al. (2018) note, this provides only a limited test of generalization, as many test examples feature a logical form that overlaps with the training data; they introduce a more challenging *query* split to ensure that neither questions nor logical forms are repeated (even after anonymizing named entities).

We extract fragments with at most 2 gaps and at most 12 tokens. On the SQL query split, the original training set contains 695 examples. GECA generates an additional 1055 using 839 distinct templates and 379 distinct fragments. For the question split we use the baseline model of Jia and Liang (2016); for the query split we use the same sequence-to-sequence model as used for SCAN and introduce the supervised copy mechanism of Finegan-Dollak et al. (2018). Environments are again taken to be identical to templates.

Results are shown in Table 2. On the split for which Jia and Liang (2016) report results, GECA achieves nearly the same improvements with weaker domain assumptions. On the remaining splits it is more accurate.

	Query	Question
Logical forms		
seq2seq	0.62 ± 0.07	0.76 ± 0.02
+ Jia et al. 16	0.61 ± 0.03	**0.81** ± 0.01
+ GECA	$\mathbf{0.65}^{\dagger\ddagger}$ ± 0.06	$0.78^{\dagger}$ ± 0.01
+ GECA + concat	0.63 ± 0.04	$0.79^{\dagger}$ ± 0.01
SQL queries		
Iyer et al. 17	0.40	0.66
seq2seq	0.39 ± 0.05	**0.68** ± 0.02
+ GECA	$\mathbf{0.49}^{\dagger}$ ± 0.02	**0.68** ± 0.02

Table 2: Meaning representation *exact-match* accuracies on the GeoQuery dataset. On logical forms, GECA approaches the data augmentation approach of Jia and Liang (2016) on the standard split of the data ("Question") and outperforms it on a split designed to test compositionality ("Query"). On SQL expressions, GECA leads to substantial improvements on the query split and achieves state-of-the-art results. Scores are averaged over 10 random seeds; the standard deviation across seeds is shown.[1] †Significant improvement over seq2seq baseline ($p < 0.01$). ‡Significant improvement over Jia and Liang (2016) ($p < 0.001$). (A t-test is used for LF experiments and a paired binomial test for SQL.)

Analysis: examples Synthesized examples for the logical and SQL representations are shown in Figure 4. Despite the fact that the sequence-to-sequence model uses neither gold entities or specialized entity linking machinery, the augmentation procedure successfully aligns natural language entity names to their logical representations and generalizes across entity choices. This procedure also produces plausible but unattested entities like a river named *florida* and a state named *west wyoming*.

The last example in the "logical forms" section is particularly interesting. The extracted fragment contains *lowest population density* on the natural language side but only `density` on the logical form side. However, the *environment* constrains substitution to happen where appropriate: this fragment will only be used in cases where the environment already contains the necessary `smallest`.

Some substitutions are semantically problematic: for example, the final datapoint in Figure 4 asks about the population of a number (because substitution has replaced *capital* with *area*); the corresponding SQL expression would fail to execute. Aside from typing problems, however, the example is syntactically well-formed and provides correct evidence about constituent boundaries, alignments and hierarchical structure within the geography domain. Other synthesized examples (like the second-to-last in Figure 4) have correct meaning representations but ungrammatical natural language inputs.

[1] In some cases these averages are slightly lower than the single-run results previously reported in the literature. Note also that the original publication from Jia and Liang reports *denotation* accuracies; the results here are taken from their accompanying code release. Overall trends across systems are comparable using either evaluation metric.

Logical forms

what is the lowest point in rhode island

```
( A , lowest ( A , ( place ( A ) , loc ( A , B ) , const ( B , stateid ( rhode island ) ) ) ) )
```

what states does the florida run through

```
( A , ( state ( A ) , const ( B , riverid ( florida ) ) , traverse ( B , A ) ) )
```

what state borders the state with the lowest population density

```
( A , ( state ( A ) , next_to ( A , B ) , smallest ( C , ( state ( B ) , density ( B , C ) ) ) ) )
```

SQL queries

what rivers run through west wyoming

```
SELECT RIVER0.NAME FROM RIVER AS RIVER0 WHERE RIVER0.TRAVERSE = " west wyoming "
```

which states have towns major named springfield

```
SELECT CITY0.STATE_NAME FROM CITY AS CITY0 WHERE CITY0.NAME = " springfield " AND CITY0.POP > 150000
```

what is the population of the area of the largest state

```
SELECT CITY0.POP FROM CITY AS CITY0 WHERE CITY0.NAME = ( SELECT STATE0.AREA FROM STATE AS STATE0
WHERE STATE0.AREA = ( SELECT MAX ( STATE1.AREA ) FROM STATE AS STATE1 ) )
```

Figure 4: Examples synthesized for semantic parsing on GEOQUERY. Substituted fragments are underlined. GECA aligns named entities to their logical representations and abstracts over predicates. Sometimes (as in the final example) synthesized examples are semantically questionable but have plausible hierarchical structure.

Analysis: dataset statistics Applying GECA to the GEOQUERY data increases full example overlap (described at the end of Section 4) by 5% for the question split in both languages, 6% for the query split with logical forms, and 9% for the query split with SQL expressions, in line with the observation that accuracy improvements are greater for the query split than the question split. Augmentation increases token co-occurrence overlap by 3–4% across all conditions.

In a larger-scale manual analysis of 100 synthesized examples from the query split, evaluating them for **grammaticality** and **accuracy** (whether the natural language captures the semantics of the logical form), we find that 96% are grammatical, and 98% are semantically accurate.

Negative results We conclude with a corresponding set of experiments on the SCHOLAR text-to-SQL dataset of Iyer et al. (2017), which is similar to GEOQUERY in size, diversity and complexity. In contrast to GEOQUERY, however, application of GECA to SCHOLAR provides no improvement. On the query split, there is limited compositional re-use of SQL sub-queries (in line with the observation of Finegan-Dollak et al. (2018) that average nesting depth in SCHOLAR is roughly half that of GEOQUERY). Correspondingly, full example overlap after augmentation remains at 0% and token co-occurrence overlap increases by only 1%. On the question split, full example overlap is larger (8%) but token co-occurrence overlap increases by less than 1%. These results suggest that GECA is most successful when it can increase similarity of word co-occurrence statistics in the training and test sets, and when the input dataset exhibits a high degree of recursion.

	Query	Question
SQL queries		
seq2seq	0.03 ± 0.01	0.57 ± 0.02
+ GECA	0.03 ± 0.01	0.56 ± 0.02

Table 3: Negative results: meaning representation accuracies on the SCHOLAR dataset. For the query split, synthesized examples do not overlap with any of the held-out data; for the question split, they provide little information beyond what is already present in the training dataset. In both cases a model trained with GECA performs indistinguishably from a the baseline model.

6 Low-resource language modeling

Both of the previous sections investigated conditional models. The fragments extracted and reused by GECA were essentially synchronous lexicon entries, in line with example (4). We originally motivated GECA with monolingual problems in which we simply wish to improve model judgments about well-formedness, so we conclude with a set of language modeling experiments.

We use Wikipedia dumps[2] in five languages (Kinyarwanda, Lao, Pashto, Tok Pisin, and a subset of English Wikipedia) as well as the Na dataset of Adams et al. (2017). These languages exhibit the performance of GECA across a range of morpholog-

[2] https://dumps.wikimedia.org/

	ENG	KIN	LAO	NA	PUS	TOK
# train tokens	2M	62K	10K	28K	2M	30K
5-MKN	369	241	315	**45.4**	574	**44.3**
+ GECA	**365**†	**239**†	**313**†	**45.4**	**570**†	**44.1**

Table 4: Perplexities on low-resource language modeling in English (ENG), Kinyarwanda (KIN), Lao, Na, Pashto (PUS) and Tok Pisin (TOK). Even with a Kneser–Ney smoothed 5-gram model (5-MKN) rather than a high-capacity neural model, applying GECA leads to small improvements in perplexity. †Significant improvement over 5-gram MKN baseline (paired binomial test, $p < 0.05$).

ical complexities: for example, Kinyarwanda has a complex noun class system (Kimenyi, 1980) and Pashto has rich derivational morphology (Tegey and Robson, 1996), while Lao and Tok Pisin are comparatively simple morphologically (Enfield, 2008; Verhaar, 1995). Training datasets range from 10K–2M tokens. Like Adams et al., we found that a 5-gram modified Kneser–Ney language model (Ney et al., 1994) outperformed several varieties of RNN language model, so we base our GECA experiments on the n-gram model instead. We use the implementation provided in KenLM (Heafield, 2011).

We extract fragments with no gaps and a maximum size of 2 tokens, with the environment taken to be a 2-token window around the extracted fragment. New usages are generated only for fragments that occur fewer than 20 times in the data. In Kinyarwanda, the base dataset contains 3358 sentences. GECA generates an additional 913, using 913 distinct templates and 199 distinct fragments.

Rather than training directly on the augmented dataset, as in preceding sections, we found that the best performance came from training one language model on the original dataset and one on the augmented dataset, then interpolating their final probabilities. The weight for this interpolation is determined on a validation dataset and chosen to be one of 0.05, 0.1 and 0.5.

Results are shown in Table 4. Improvements are not universal and are more modest than in preceding sections. However, GECA decreases perplexities across multiple languages and never increases them. These results suggest that the substitution principle underlying GECA is a useful mechanism for encouraging compositionality even outside conditional tasks and neural models.

Analysis: examples and statistics In language modeling, GECA functions as a smoothing scheme: its primary effect is to move mass toward n-grams that can appear in productive contexts. In this sense, GECA performs a similar role to the Kneser–Ney smoothing also used in all LM experiments. With GECA, in contrast to Kneser–Ney, the notion of "context" can look forward as well as backward, and capture longer-range interactions.

Examples of synthesized sentences are shown in Figure 5. Most sentences are grammatical, and many of the substitutions preserve relevant semantic type information (substituting locations for locations, times for times, etc.). However, some ill-formed sentences are also generated.

As in Section 5, we manually inspect 100 synthesized sentences. As before, sentences are evaluated for grammaticality; here, since no explicit semantics were provided, they are instead evaluated for generic semantic acceptability. In this case, only 51% of synthesized sentences are semantically acceptable, but 79% are grammatical.

7 Discussion

We introduced GECA, a simple data augmentation scheme based on identifying local phrase substitutions that are licensed by common contexts, and demonstrated that extra training examples generated with GECA lead to substantial improvements on both diagnostic and natural datasets for semantic

various copies of portions of the code of hammurabi have been found on baked clay tablets , some possibly older than the celebrated basalt stele now in the night sky .
the work contains , in an appendix , the german equivalents for the technical terms used in the glock $num .
payments system in the aclu proposed new directions for the organization .
in the late triassic and early nineteenth century , a number of scots-irish traders lived among the choctaw and married high-status women .

Figure 5: Sentences synthesized for the English language modeling task. Most examples are syntactically well-formed; some are also semantically plausible.

parsing and language modeling.

While the approach is surprisingly effective in its current form, we view these results primarily as an invitation to consider more carefully the role played by representations of sentence fragments in larger questions about compositionality in black-box sequence models. The procedure detailed in this paper relies on exact string matching to identify common context; future work might take advantage of learned representations of spans and their environments (Mikolov et al., 2013; Peters et al., 2018). Further improvements are likely obtainable by constraining the extracted fragments to respect constituent boundaries when syntactic information is available.

The experiments presented here focus on rewriting sentences using evidence *within* a dataset to encourage generalization to new outputs. An alternative line of work on paraphrase-based data augmentation (Ganitkevitch et al., 2013; Iyyer et al., 2018) uses external, text-only resources to encourage robust interpretation of new inputs corresponding to known outputs. The two lines of work could be combined, e.g. by using GECA-identified fragments to indicate productive locations for sub-sentential paraphrasing.

More generally, the present results underline the extent to which current models fail to learn simple, context-independent notions of reuse, but also how easy it is to make progress towards addressing this problem without fundamental changes in model architecture.

Reproducibility

Code for all experiments in this paper may be found at `github.com/jacobandreas/geca`.

Acknowledgments

Thanks to Oliver Adams for assistance with the language modeling experiments, and to the anonymous reviewers for suggestions in the analysis sections.

References

Oliver Adams, Adam Makarucha, Graham Neubig, Steven Bird, and Trevor Cohn. 2017. Cross-lingual word embeddings for low-resource language modeling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Joost Bastings, Marco Baroni, Jason Weston, Kyunghyun Cho, and Douwe Kiela. 2018. Jump to better conclusions: Scan both left and right. In *Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Jean Berko. 1958. The child's learning of English morphology. *Word*.

Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*.

Ciprian Chelba and Frederick Jelinek. 1998. Exploiting syntactic structure for language modeling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 263–270, Ann Arbor, Michigan.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT press.

Alexander Clark. 2000. Inducing syntactic categories by context distribution clustering. In *Conference on Natural Language Learning*.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. 2016. Recurrent neural network grammars. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics*.

Nicholas J Enfield. 2008. *A grammar of Lao*, volume 38. Walter de Gruyter.

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 Annual Meeting of the Association for Computational Linguistics*.

John Rupert Firth. 1957. *A Synopsis of Linguistic Theory, 1930–1955*.

Kuzman Ganchev, Jennifer Gillenwater, Ben Taskar, et al. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics*.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural Turing machines. *arXiv preprint arXiv:1410.5401*.

Edward Grefenstette, Karl Moritz Hermann, Mustafa Suleyman, and Phil Blunsom. 2015. Learning to transduce with unbounded memory. In *Advances in Neural Information Processing Systems*.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Workshop on Machine Translation*.

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

Zhiting Hu, Bowen Tan, Ruslan Salakhutdinov, Tom Mitchell, and Eric P Xing. 2019. Learning data manipulation for augmentation and weighting. *arXiv preprint arXiv:1910.12795*.

Zhiting Hu, Zichao Yang, Ruslan R Salakhutdinov, Lianhui Qin, Xiaodan Liang, Haoye Dong, and Eric P Xing. 2018. Deep generative models with learnable knowledge constraints. In *Advances in Neural Information Processing Systems*.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. 2017. Learning a neural semantic parser from user feedback. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics*.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Alexandre Kimenyi. 1980. *A relational grammar of Kinyarwanda*, volume 91. Univ. of California Press.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Dan Klein and Christopher D Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Kevin Knight and Jonathan Graehl. 2005. An overview of probabilistic tree transducers for natural language processing. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–24, Mexico City, Mexico.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*.

Brenden M Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the International Conference on Machine Learning*.

Brenden M Lake, Tal Linzen, and Marco Baroni. 2019. Human few-shot learning of compositional instructions. *arXiv preprint arXiv:1901.04587*.

Gábor Melis, Chris Dyer, and Phil Blunsom. 2018. On the state of the art of evaluation in neural language models. In *Proceedings of the International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Richard Montague. 1973. The proper treatment of quantification in ordinary english. In *Approaches to natural language*.

Hermann Ney, Ute Essen, and Reinhard Kneser. 1994. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech & Language*.

Junhyuk Oh, Satinder Singh, Honglak Lee, and Pushmeet Kohli. 2017. Zero-shot task generalization with multi-task deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics*.

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

Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. 2017. Learning to compose domain-specific transformations for data augmentation. In *Advances in Neural Information Processing Systems*, pages 3236–3246.

Jake Russin, Jason Jo, and Randall C O'Reilly. 2019. Compositional generalization in a deep seq2seq model by separating syntax and semantics. *arXiv preprint arXiv:1904.09708*.

Cecilia Summers and Michael J Dinneen. 2019. Improved mixed-example data augmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1262–1270. IEEE.

Ilya Sutskever, Oriol Vinyals, and Quoc Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*.

Habibullah Tegey and Barbara Robson. 1996. A reference grammar of Pashto.

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

John WM Verhaar. 1995. *Toward a reference grammar of Tok Pisin: An experiment in corpus linguistics*, volume 26. University of Hawaii Press.

Jason Wei and Kai Zhou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

John Zelle. 1995. *Using Inductive Logic Programming to Automate the Construction of Natural Language Parsers*. Ph.D. thesis, Department of Computer Sciences, The University of Texas at Austin.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 658–666.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference on Learning Representations*.