

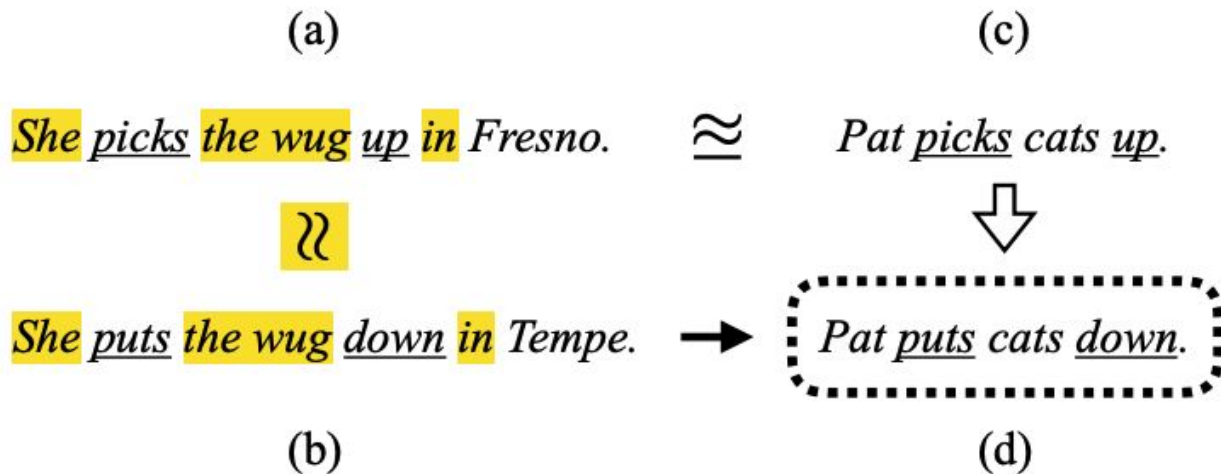


Compositional Generalization in Large LM Era

Jingfeng Yang

What is Compositional Generalization?

Compositional generalization is the ability to generalize systematically to a new data distribution by combining known components



Why is Compositional Generalization Important ?



1. Core of ML: Generalization -> Out-of-distribution Generalization -> Compositional Generalization (Language structure)
2. Still limitations of Large Language Models (LM)
3. Critical component in Reasoning (highly related to commonsense reasoning, relational reasoning etc.)
4. Compositional Generalization Beyond Language

Compositional Generalization Beyond Language

TEXT DESCRIPTION

An astronaut Teddy bears A bowl of
soup

riding a horse lounging in a tropical resort
in space playing basketball with cats in
space

in a photorealistic style in the style of Andy
Warhol as a pencil drawing



DALL-E 2



How to improve Compositional Generalizability?

Model Perspective:

Modular Networks, Intermediate Abstractions, Neural-Symbolic Models etc.

Ours:

SEQZERO: Few-shot Compositional Semantic Parsing with Sequential Prompts and Zero-shot Models

**Jingfeng Yang[†] Haoming Jiang[†] Qingyu Yin[†]
Danqing Zhang[†] Bing Yin[†] Diyi Yang[†]**

[†] Amazon

[‡] Georgia Institute of Technology

{jingfe, jhaoming, qingyy, danqinz, alexbyin}@amazon.com
dyang888@gatech.edu

How to improve Compositional Generalizability?



Data Perspective:

Data Augmentation: SUBS, GECA

Ours:

SUBS: Subtree Substitution for Compositional Semantic Parsing


Jingfeng Yang^{†*} Le Zhang^{†*} Diyi Yang[†]

[†] Georgia Institute of Technology

[†] Fudan University

`jingfengyangpku@gmail.com zhangle18@fudan.edu.cn`

`dyang888@gatech.edu`



SeqZero: Few-shot **Compositional** Semantic Parsing with **Sequential Prompts** and **Zero-shot Models**

Jingfeng Yang, Haoming Jiang, Qingyu Yin, Danqing Zhang, Bing Yin, Diyi Yang

Compositional Generalization in Semantic Parsing

Semantic Parsing: Natural Language utterance -> Formal Language utterance (e.g. SQL Query)

Training Example 1:

Natural: How many people live in Chicago ?

Formal (SQL): SELECT city.population FROM city WHERE city.city_name = "Chicago"

Training Example 2:

Natural: Give me the state that borders Utah .

Formal (SQL): SELECT border_info.border FROM border_info WHERE border_info.state_name = "Utah"

Test Example:

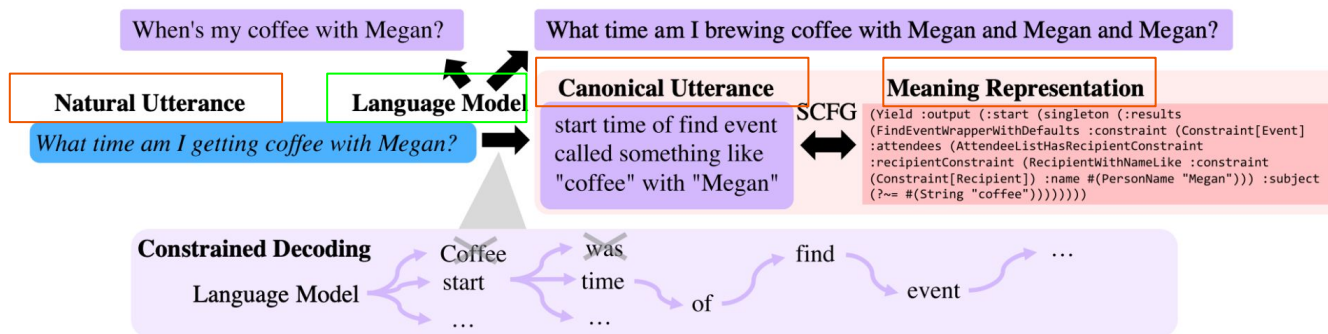
Natural: How many people live in Utah ?

Formal (FunQL): SELECT state.population FROM state WHERE state.state_name = "Utah"

Examples are from GeoQuery dataset.

Prior Work: Semantic Parsing via Paraphrasing (SPP) and LMs

- Schucher et al., 2021, Shin et al., 2021



Natural Utterance -> Canonical Utterance -> Formal Language Utterance

↑
Pretrained Language Models

↑
Rules or Grammar

Problem 1: Lengthy and Complex Output



The canonical utterance is lengthy and complex due to compositional structure of the formal languages, which is still hard for LMs

Solution: Decompose the problem into a sequence of sub-problems, and the LMs only need to make a sequence of short prompt-based predictions.

Problem 2: Spurious Biases in Compositional Generalization

Question:
how many people live in Utah ?

Gold SQL:
SELECT **state** . population FROM **state**
WHERE **state** . **state_name** = "Utah"

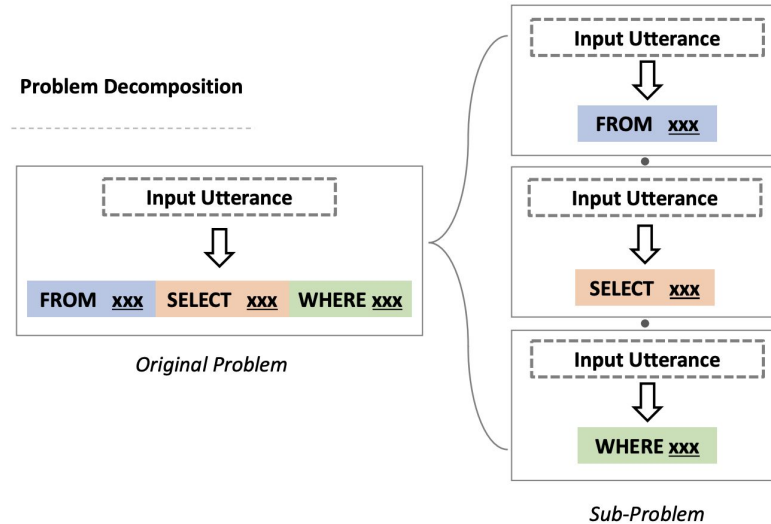
Finetuned BART Predicted SQL:
SELECT **city** . population FROM **city**
WHERE **city** . **city_name** = "Utah"

Solution:

- Ensemble of
 - Pertained models: better out-of-distribution (OOD) generalizability.
 - Fine-tuned models: better in-distribution generalizability.
- Has both advantages and avoids overfitting.

Figure 1: Finetuned BART's OOD generalization errors due to overfitting the spurious biases.

Problem Decomposition and Sequential Prompt Filling



Each sub-problem is finished by filing in a prompt by a LM.

Ensemble of Few-shot and Zero-shot Models

Constrained rescaling of zero-shot models:

Probability of zero-shot LM

Rescaled probability
of zero-shot LM

$$P_{\theta_{i,z}}(w|x) = \frac{\mathbb{1}(w \in V_i(x)) P_{\theta_0}(w|x)}{\sum_{w_j \in V_i(x)} P_{\theta_0}(w_j|x)},$$

Ensemble:

Allowed vocabulary given prefix

$$P_{\theta_i} = \gamma_i P_{\theta_{i,f}} + (1 - \gamma_i) P_{\theta_{i,z}},$$

Final probability Probability of few-shot LM

Overview of SeqZero

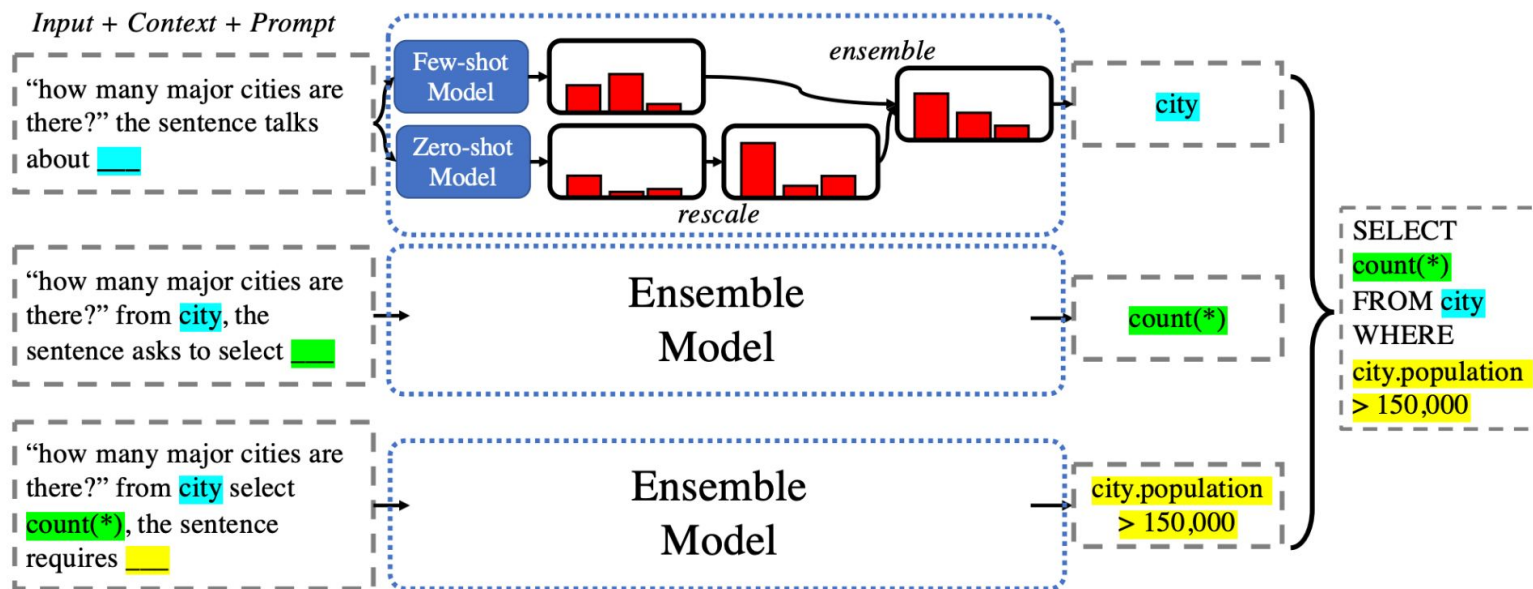


Figure 3: Pipeline of sequential prompt filling and SQL generation on GeoQuery. Note that, the scale of the prediction probability of the zero-shot model is very small before rescaling.

Dataset and Evaluation



- Dataset:
 - GeoQuery Compositional Split
 - EcommerceQuery Compositional Split

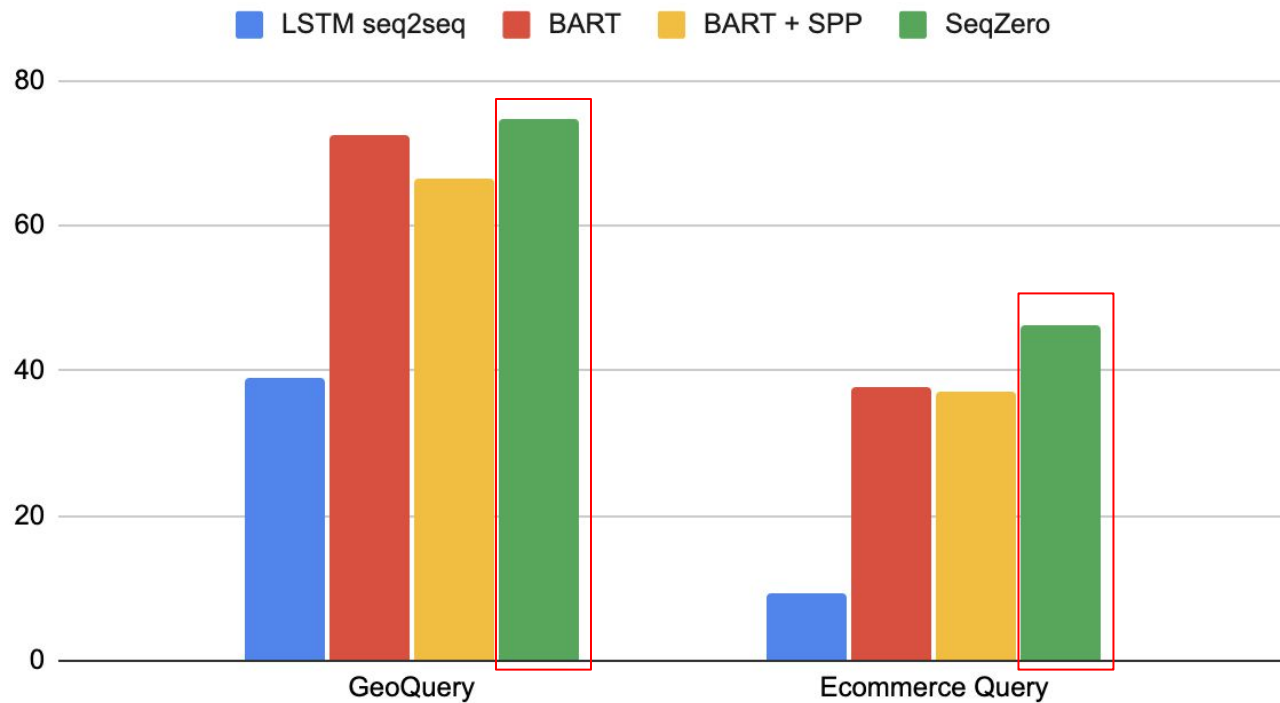
Test Example:

Natural: petrol trimmer over 100 dollar

Formal (SQL): `SELECT * FROM ASINs WHERE Maching Algorithm("petrol trimmer") == True and Price > 100`

- In training set, there are "Price <" and "Size >" combinations, but no "Price >" combination.
- Evaluation Metric:
 - Exact Match (Whole SQL utterance accuracy)

SeqZero Outperforms all Baselines



Effect of Zero-shot Models and Sequential Prompts

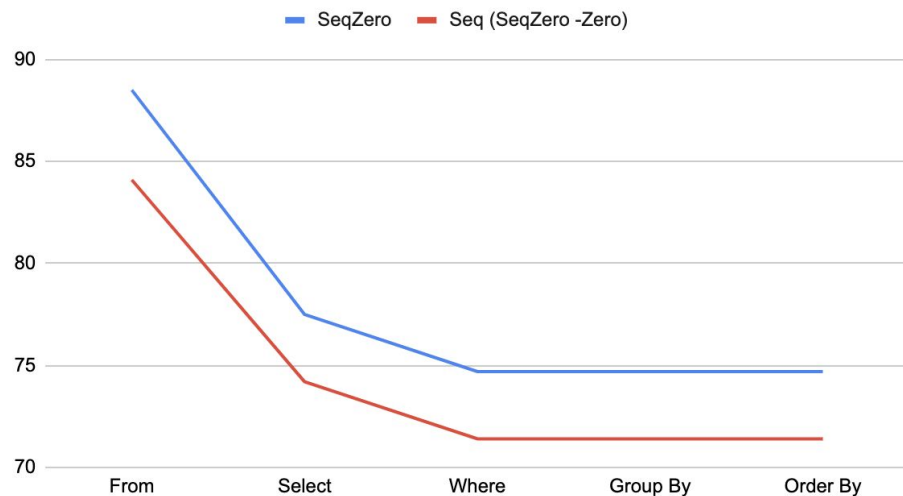


Method	GeoQuery	EcoQuery
SEQZERO	74.7	46.2
–SEQ	74.2	44.5
–ZERO	71.4	37.7

Table 2: Ablation study of SEQZERO.

- Without the help of zero-shot models, the performance decreases a lot.
- Without sequential prompts, it's hard to design specific prompts for subproblems and mine knowledge from zero-shot (pretrained) models.

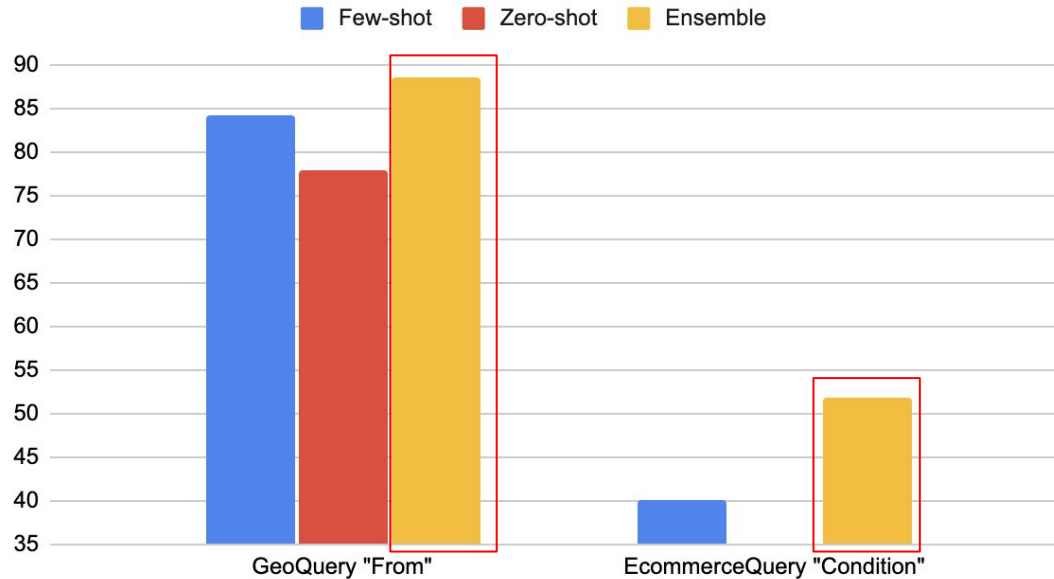
Analysis of Sequential Prompt Based Models



Ensemble of Zero-shot model in SeqZero boosts performance on the “FROM” clause, thus significantly reduces the error propagation, leading to better performance on all clauses.

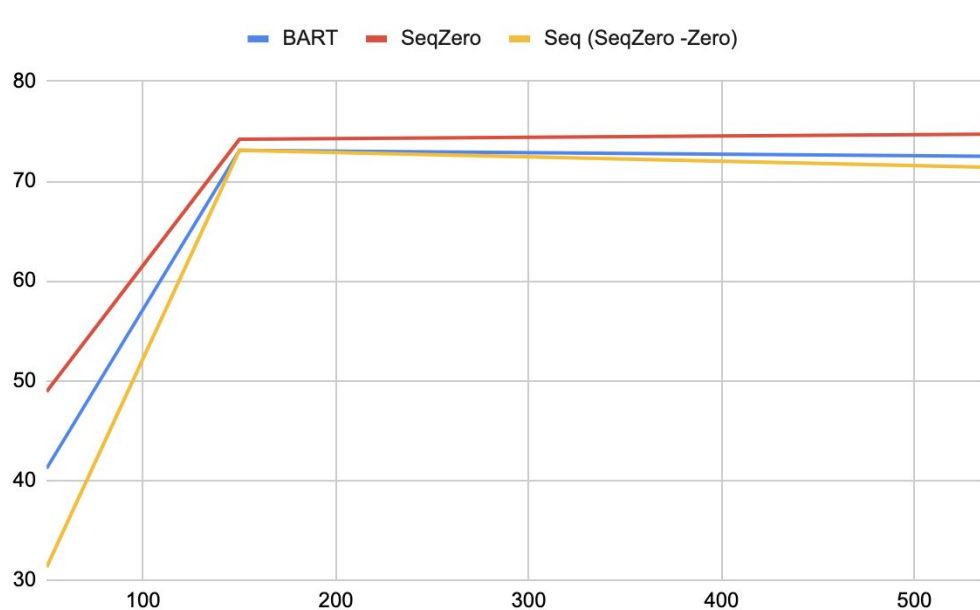
Zero-shot, Few-shot models, and Their Ensemble

Zero-shot models requires prefix constrained decoding.



Ensemble of Zero-shot (Pretrained) and Few-shot (Finetuned) models has better performance because it achieves much better compositionally OOD generalization while maintaining in-distribution generalizability.

Few-shot Settings



Before certain point, SeqZero has larger improvement with more examples. Increasing training examples with the same templates enhances overfitting of seq2seq models, leading to larger gap between SeqZero and others.

SeqZero



- Takeaways:
 - Problem decomposition and sequential prompts enables flexible prompt designing.
 - Ensemble of zero-shot (pretrained) and few-shot (finetuned) models achieves better compositional OOD generalizability, while maintaining in-distribution generalizability.
 - Constrained rescaling is important for ensemble of zero-shot and few-shot models to work in the generation task.



SUBS: Subtree Substitution for **Compositional** Semantic Parsing

Jingfeng Yang, Le Zhang, Diyi Yang

Compositional Generalization in Semantic Parsing

Training Example 1:

Natural: What is the largest city in the smallest state in the USA ?

Formal (FunQL): answer (largest (city (loc_2 (smallest (state (loc_2 (countryid (usa))))))))))))

Training Example 2:

Natural: What is the population of the largest state ?

Formal (FunQL): answer (population_1 (largest (state (all)))))

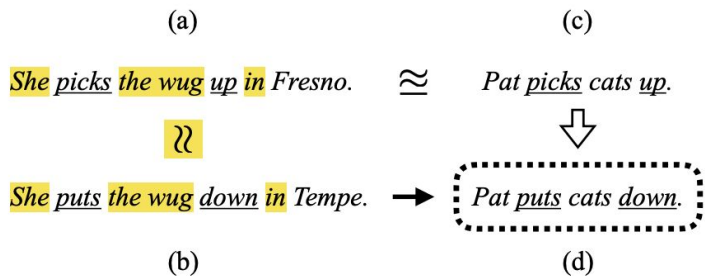
Test Example:

Natural: What is the population of the largest city in the smallest state in the USA ?

Formal (FunQL): answer (population_1 (largest (city (loc_2 (smallest (state (loc_2 (countryid (usa)))))))))))))

Prior Work for Compositional Semantic Parsing

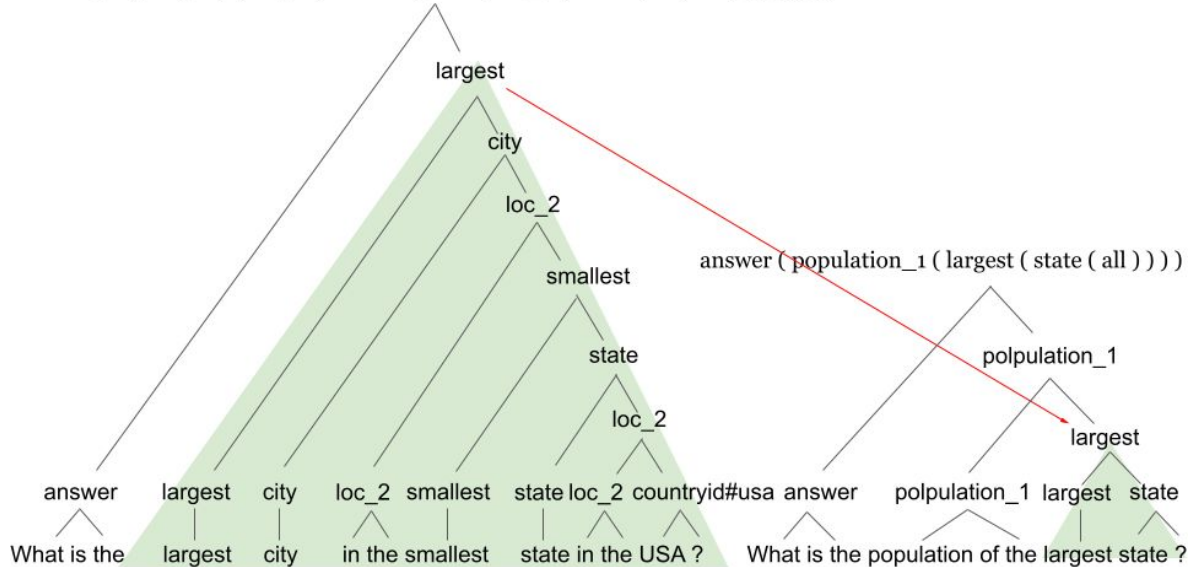
- Model Biases: Span-based Semantic Parsing (Herzig et al., 2021), Neural-Symbolic Stack Machines (Chen et al., 2020), Neural Module Networks (Gupta et al., 2019) etc.
- Data Augmentation and then Seq2seq Model:
 - Synchronous Context-Free Grammar (SCFG) (Jia et al., 2016).
 - Good-Enough Compositional Data Augmentation (GECA) (Andreas et al., 2019):



Limitations of prior Data Augmentation: identify only simple replaceable spans!

Subtree Substitution (SUBS) Data Augmentation

```
answer ( largest ( city ( loc_2 ( smallest ( state ( loc_2 ( countryid ( usa ) ) ) ) ) ) ) ) ) ) ) )
```



Subtree Substitution Result:

What is the population of the largest city in the smallest state in the USA ?

```
answer ( population_1 ( largest ( city ( loc_2 ( smallest ( state ( loc_2 ( countryid ( usa ) ) ) ) ) ) ) ) ) ) )
```


Results - SCAN



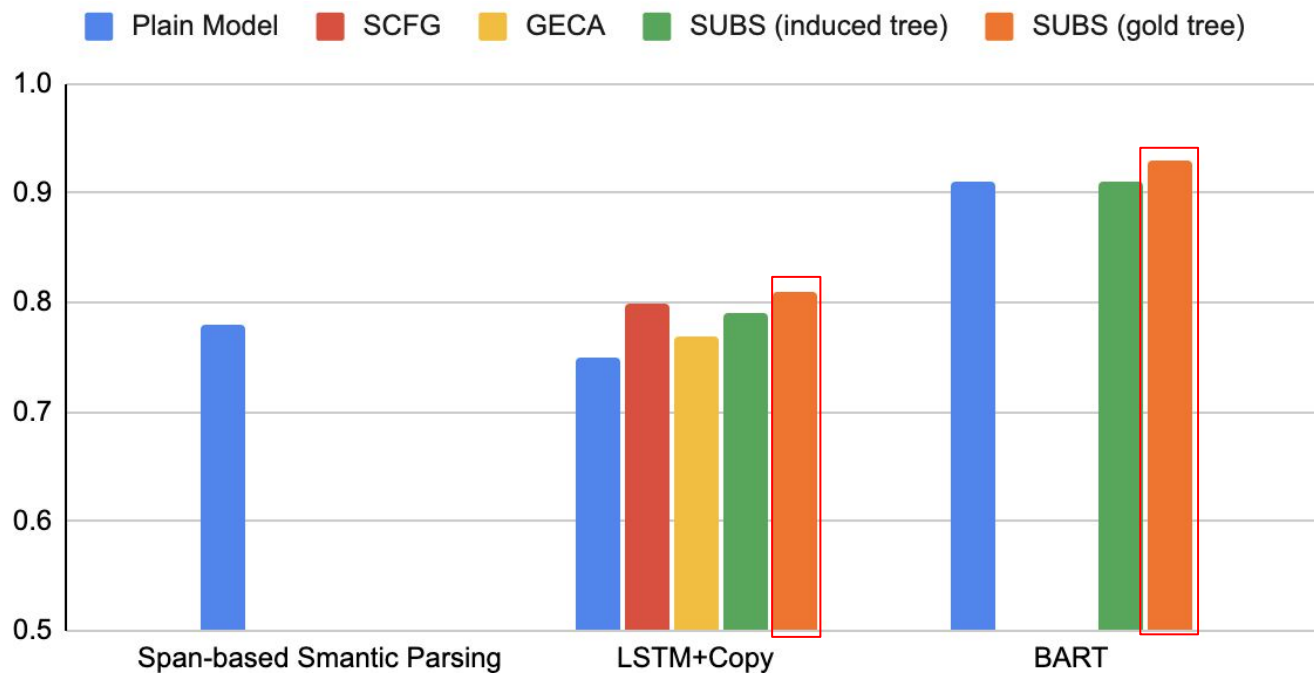
Better performance and faster convergence on the diagnostic dataset.

	RIGHT	AROUNDRIGHT
LSTM	0.00	1.00 (2800 updates)
LSTM + SUBS	1.00	1.00 (800 updates)

Table 1: Accuracy of diagnostic experiments on SCAN.

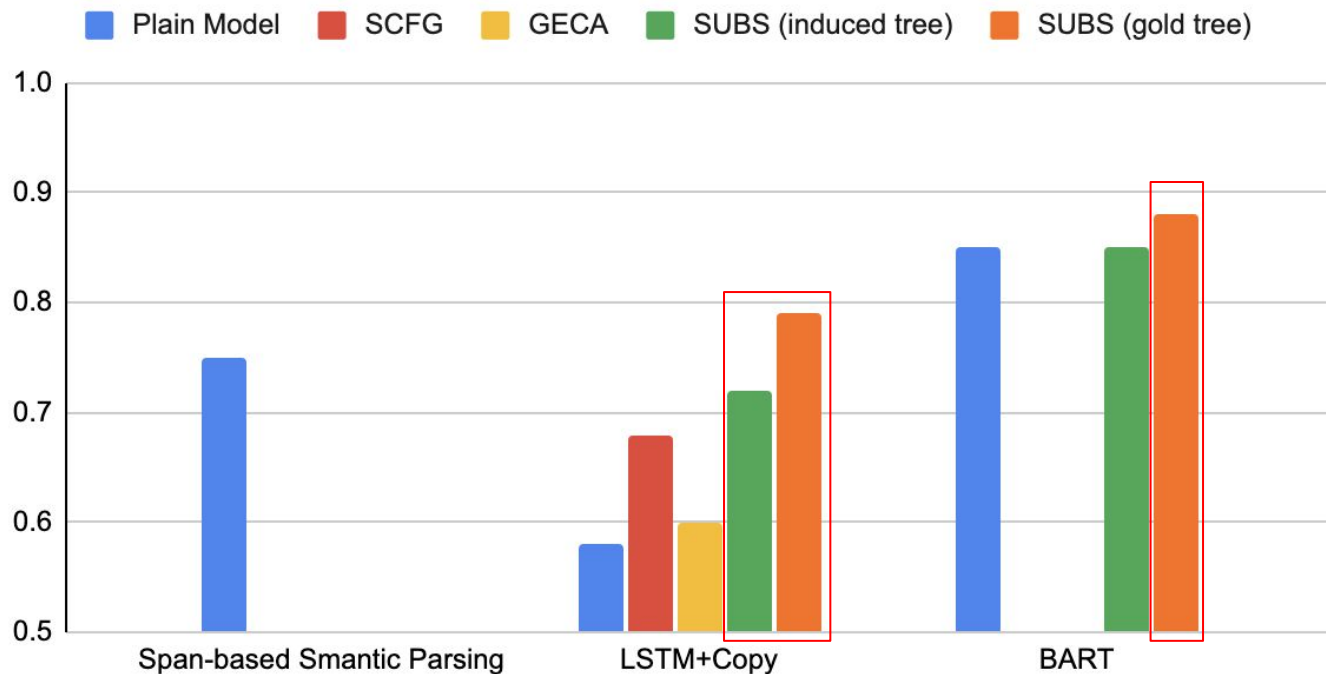
Results - GeoQuery i.i.d. Split

Data augmentation boost the performance , especially in LSTM based models.



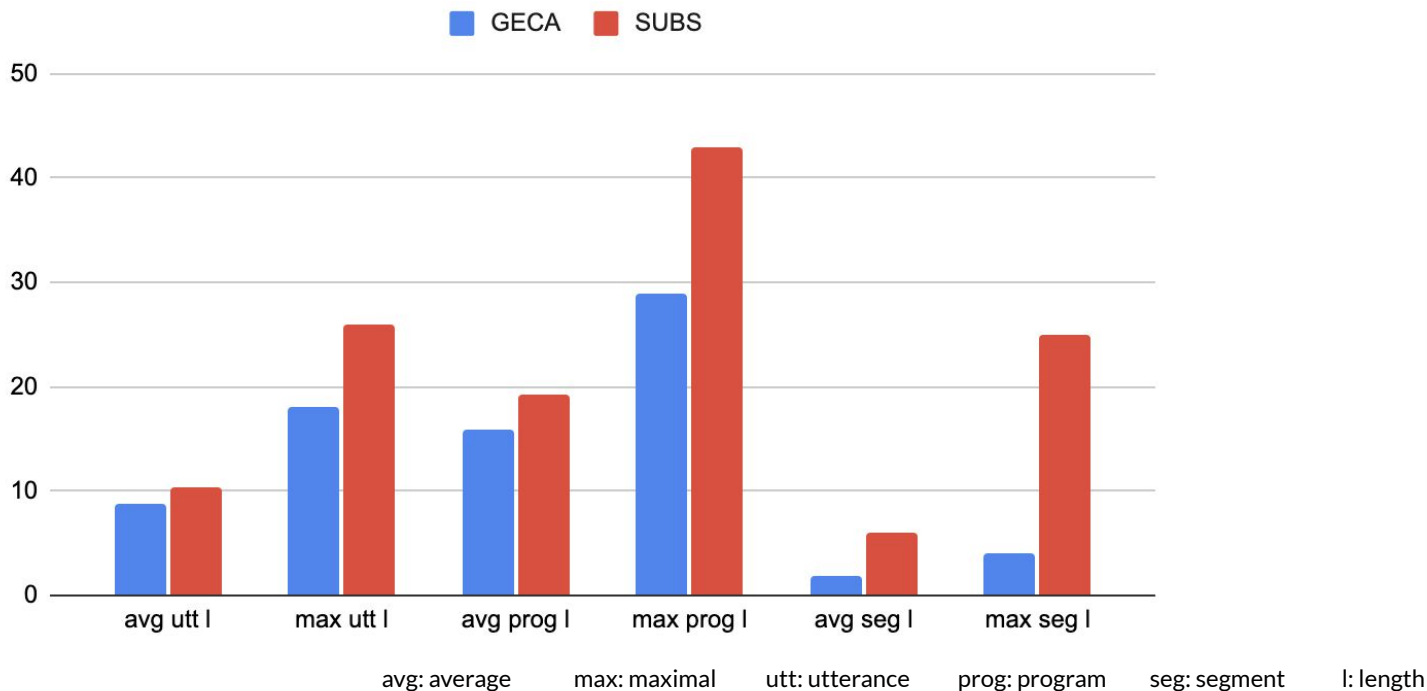
Results - GeoQuery Compositional Split

SUBS data augmentation is better than others for compositional generalization!



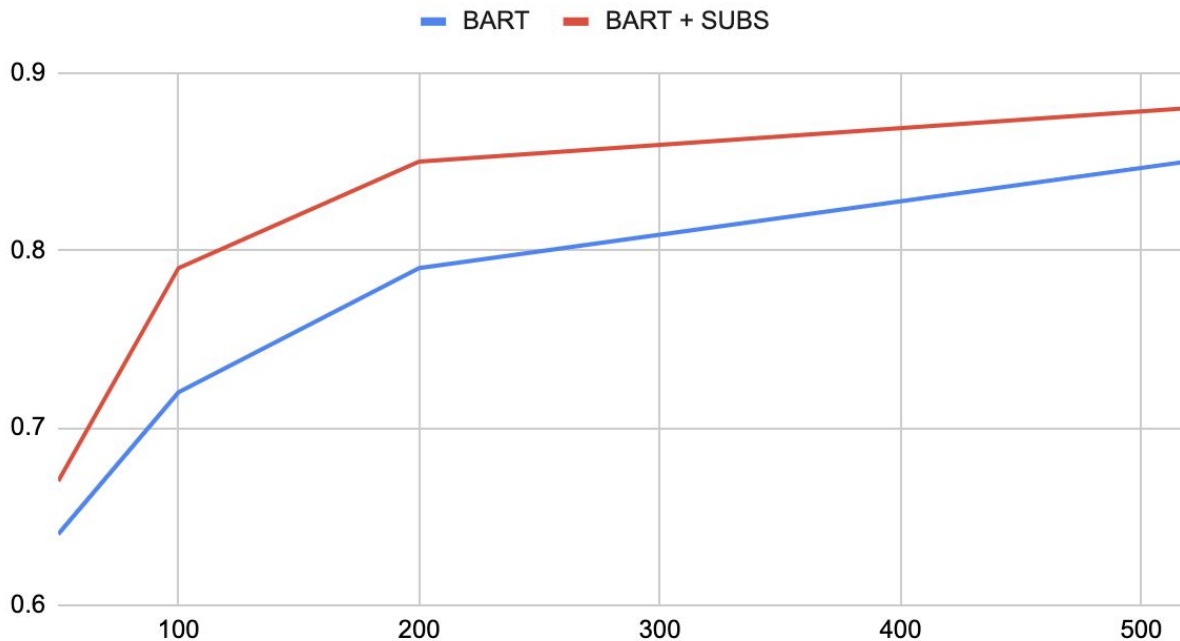
Analysis of Augmented Data

Compared with GECA, SUBS can identify and exchange much more complex structures, and produce more complex utterance and program pairs.



Few-shot Settings

The improvement of SUBS is even larger in the few-shot setting!



SUBS: Subtree Substitution (for Compositional Semantic Parsing)



- Takeaways:
 - Subtree Substitution as a Compositional data augmentation method can help compositional generalization in semantic parsing.
 - Subtree Substitution can identify more complex structures as exchangeable elements, compared with other augmentation methods.



More Recent Work

In-context Learning v.s. Fine-tuning v.s. Prompt Tuning

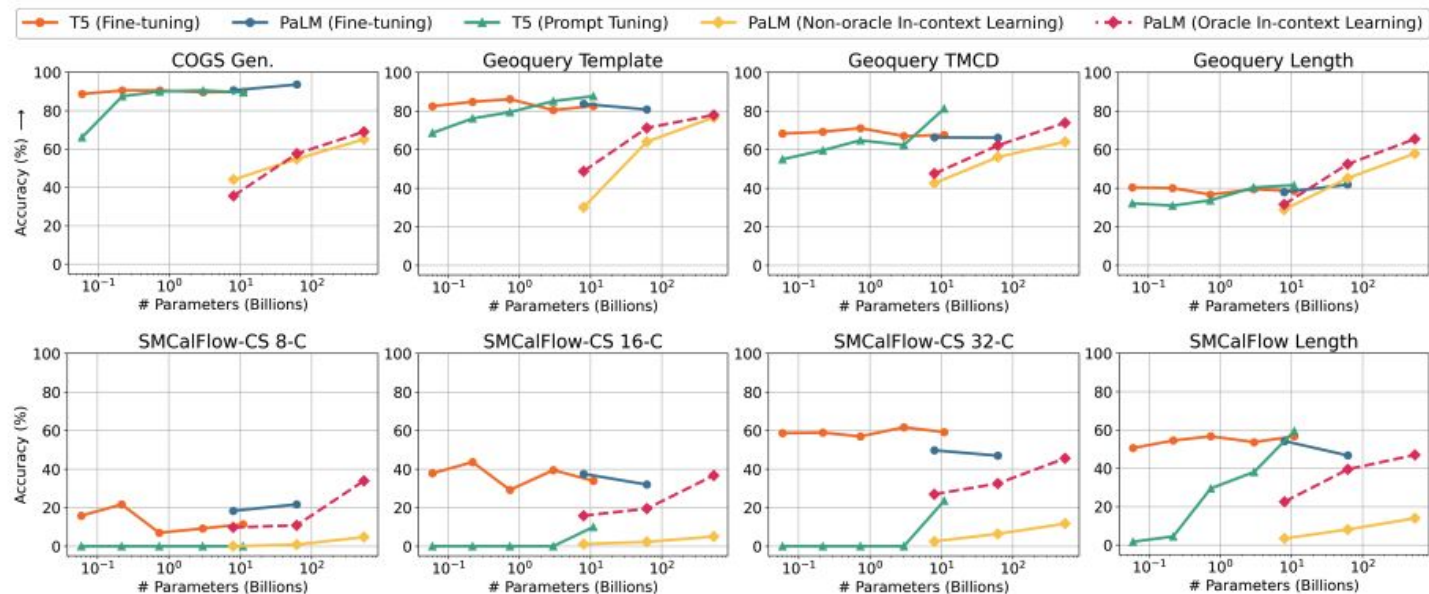


Figure 2: Scaling curves for different datasets and splits using different training schemes. Note that the in-context learning with an oracle retriever (dashed) cannot be compared directly with other methods as it has access to the gold output.

Chain-of-Thought Prompting, Least-to-Most Prompting



Prompting method	code-davinci-002	code-davinci-001	text-davinci-002*
Standard prompting	16.7	0.4	6.0
Chain-of-Thought	16.2	0.0	0.0
Least-to-Most	99.7	60.7	76.0

Table 9: Accuracies (%) of different prompting methods on the test set of SCAN under the length-based split. The results of `text-davinci-002` are based on a random subset of 100 commands.

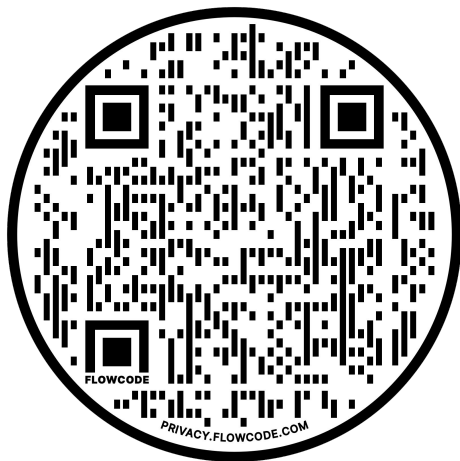
Wei J, Wang X, Schuurmans D, et al. Chain of thought prompting elicits reasoning in large language models[J]. arXiv preprint arXiv:2201.11903, 2022.

Zhou D, Schärli N, Hou L, et al. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models[J]. arXiv preprint arXiv:2205.10625, 2022.

Question:

- 1) Incorporate (structural) inductive biases directly to LMs. 2) First incorporate inductive biases to data (e.g. via data augmentation) and then used the data to help naive (seq2seq) model to have such inductive biases. Which is better?
- Are inductive biases still useful in the future with even larger models?

Compositional
Generalization
组合泛化中文
介绍:



以LM视角看
NLP热点:

