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 Genarlization 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”

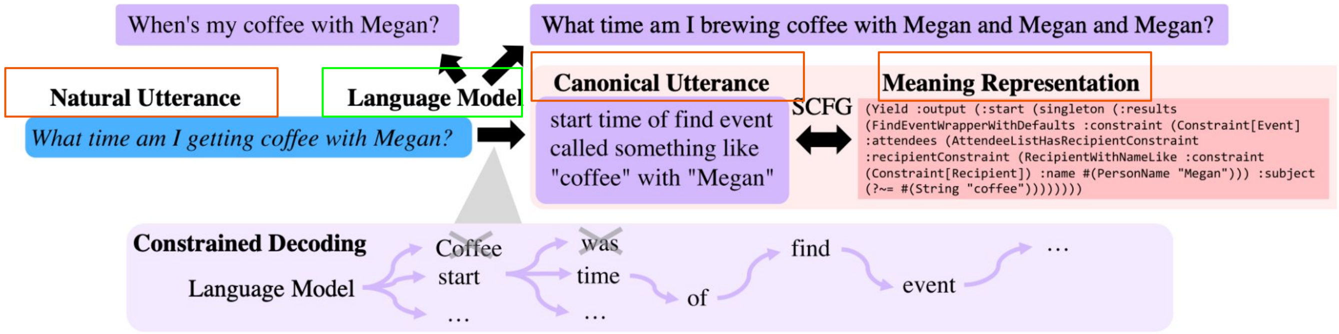
Training Example 2:
Natural: Give me the state that borders Utah .
Formal (SQL): SELECT border_info.border FROM border_info WHERE boder_info.state_name = “Utah”

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:

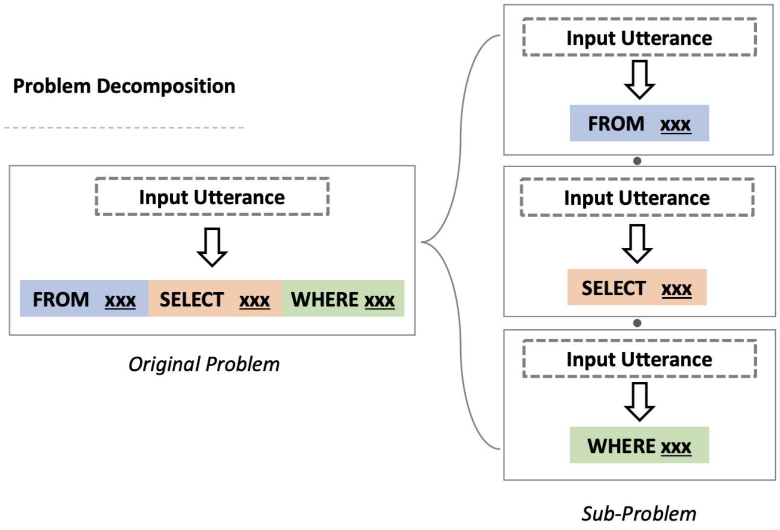
SELECT **city** . population FROM **city**
WHERE **city** . **city**_name = ”Utah"

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

Solution:

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

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

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

Allowed vocabulary given prefix

Ensemble:

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

Final probability

Probability of few-shot LM

Overview of SeqZero

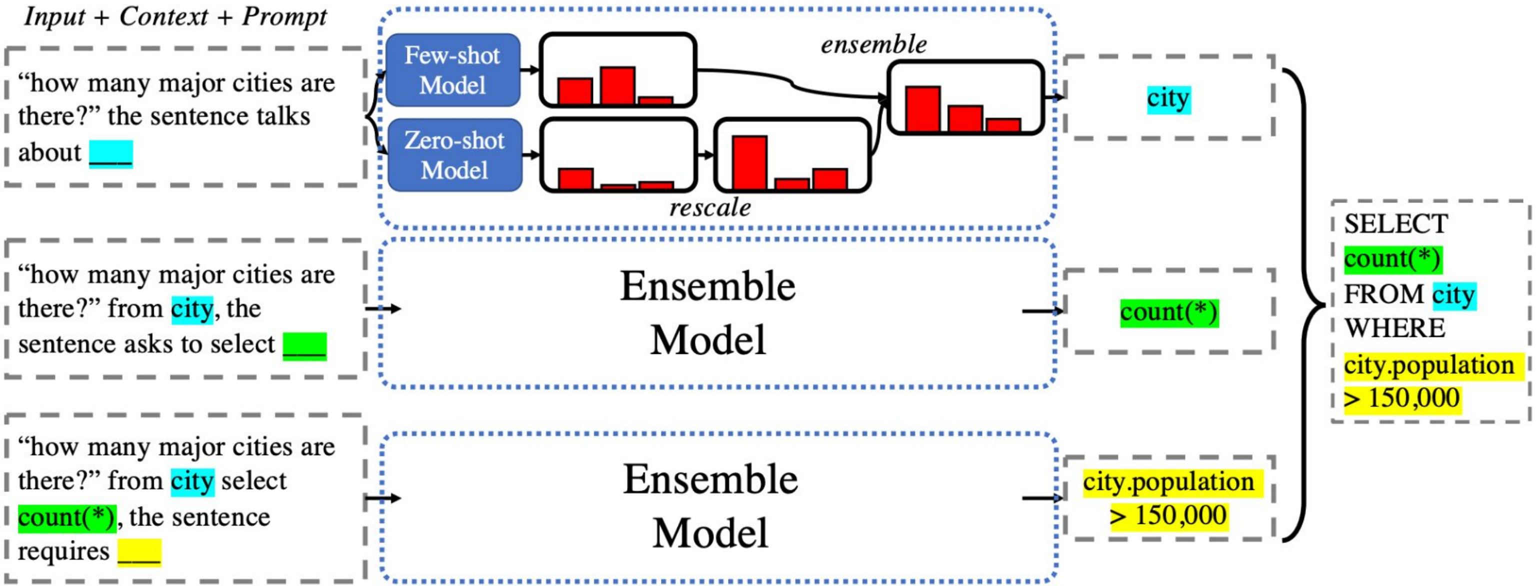


Figure 3: Pipeline of sequential prompt filling and SQL generation on GeoQuery. Note that, the scale of the prediction probability of the zero-shot model is very small before rescaling.

Dataset and Evaluation

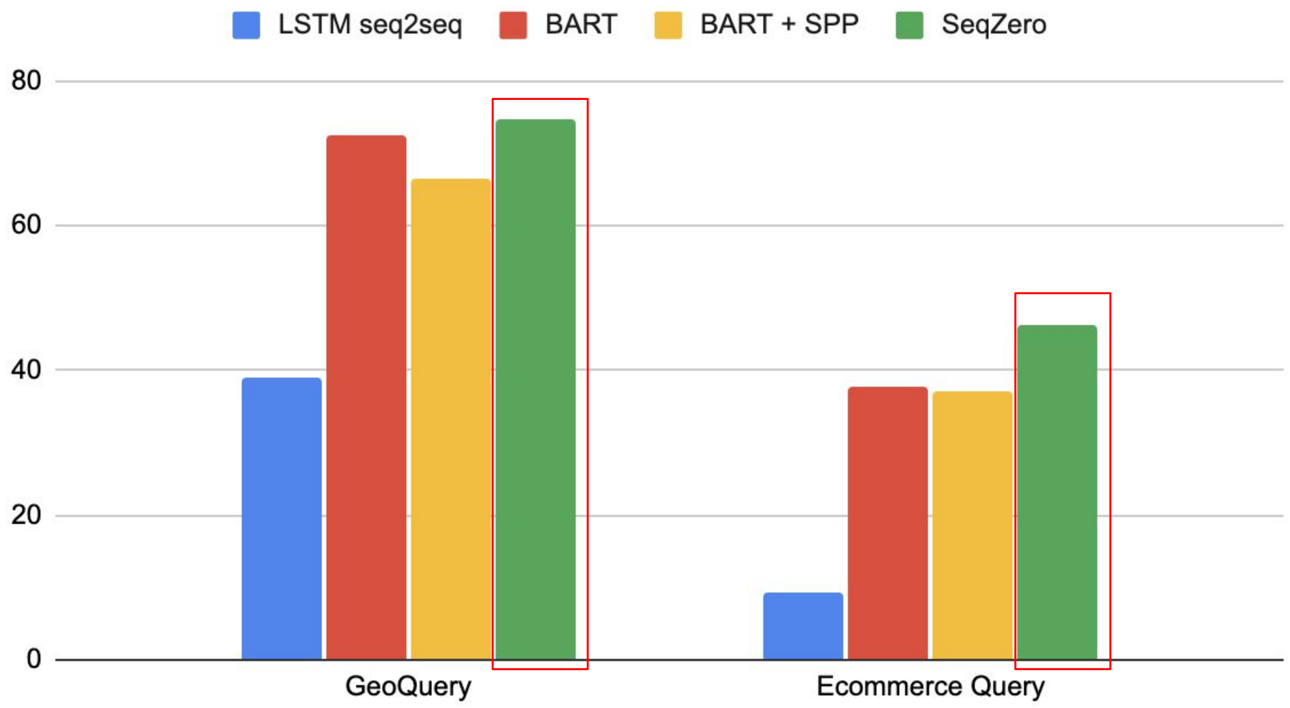
- Dataset:
  - GeoQuery Compositional Split
  - EcommerceQuery Compositional Split

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



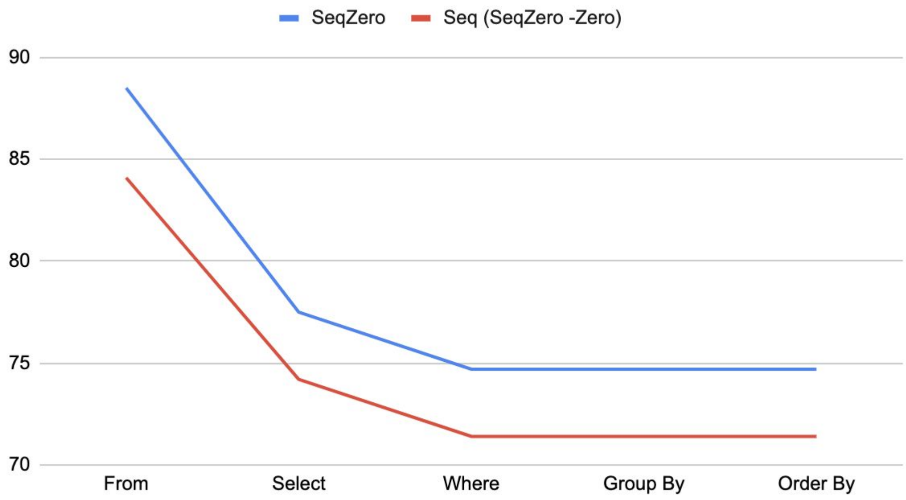
Effect of Zero-shot Models and Sequential Prompts

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.

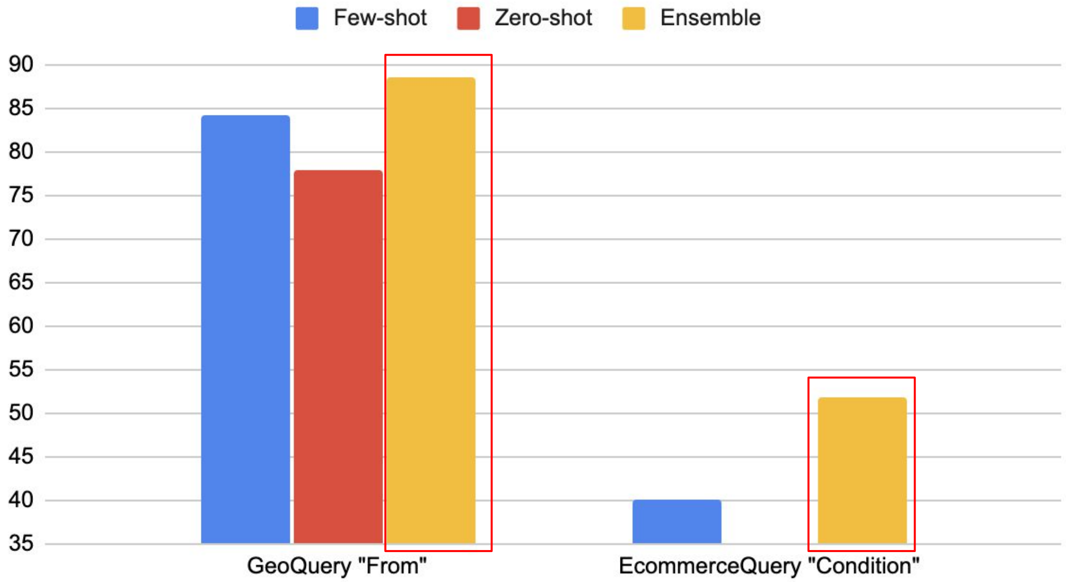
Analysis of Sequential Prompt Based Models



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.

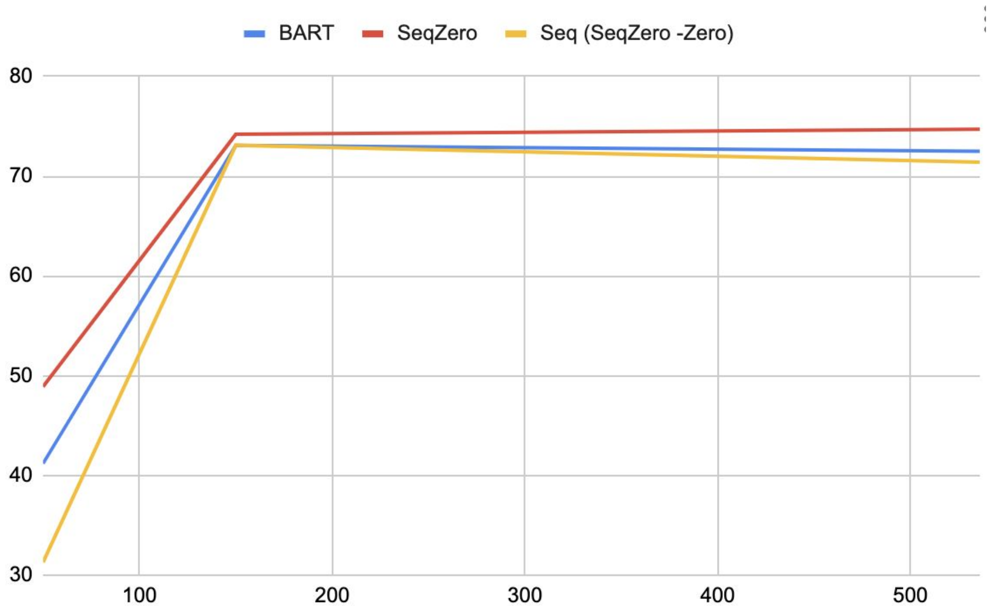
Zero-shot, Few-shot models, and Their Ensemble

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.
- Authors:
    - Jingfeng Yang, Haoming Jiang, Qingyu Yin, Danqing Zhang, Bing Yin, Diyi Yang