NLP Reading Group series: **Reasoning Like Program Executors**

NLP Reading Group, University of Arizona

Sushma Akoju
Advisor: Prof. Mihai Surdeanu

[NLP reading Group Spring 2023: Reasoning Like Program Executors]

What does Program Executor (POET) do?

Input: Program and program context

Output: Result

Claim: POET empowers LMs to harvest reasoning knowledge

- First the authors pre-train a model for Math problem
- Then pre-train another model for the Logic problems
- They use BART & Roberta as base models
- In both of above cases, they consider 3 things as input
  a. Program (conclusion/hypothesis),
  b. Program context,
  c. And result
- They follow same format for
  a. math problems,
  b. first order logic statements
  c. SQL queries.
- Fine tune on downstream tasks

https://arxiv.org/pdf/2201.11473.pdf

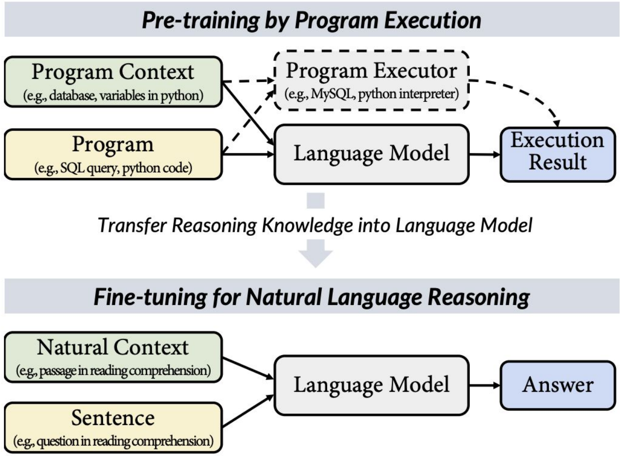
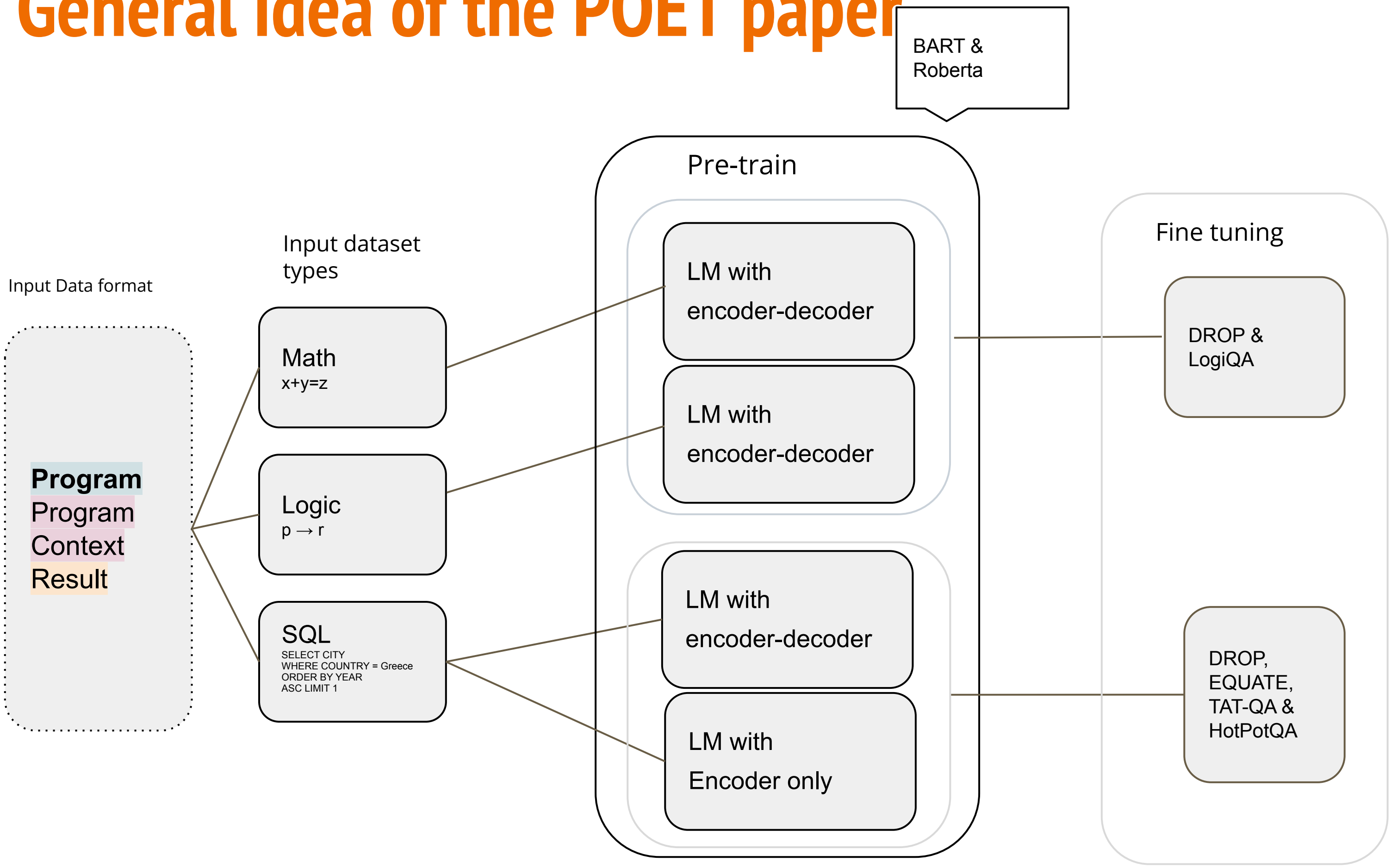


Figure 1: Given a program context and a program as input, POET pre-trains LMs to output the execution result. After fine-tuning on downstream tasks, POET can boost LMs on reasoning-required scenarios. Explanations about program context, program, program executor and execution result can be found in § 3. More examples of natural context and sentence are in Table 1.

Attempting to simplify POET's overall idea

- Can a "context-free" language corpus with program, context during pre-training of a Language Model (LM) help to reason over quantitative information in Natural Language after finetuning an LM?

General idea of the POET paper



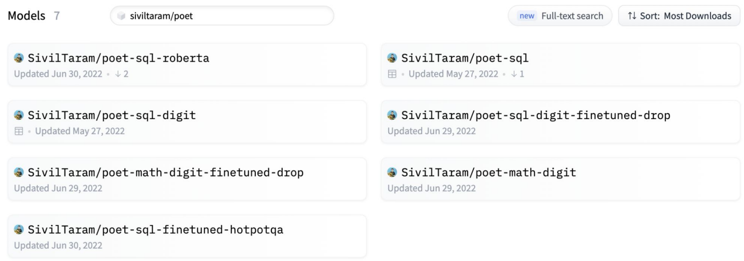
Claims

Will POET be affected by naturalness of program context or program? No.

Does pre-training on NL reasoning benefit model learning on program execution? Yes.

Can POET boost reasoning abilities of giant pre-trained language models? Yes.

HuggingFace POET: https://huggingface.co/models?search=siviltaram/poet



Datasets

Type	Example	Dataset	Task
Numerical	**Question:** What is the difference in casualty numbers between Bavarian and Austrian? **Passage:** [DOC] The popular uprising included large areas of . . .	DROP (Dua et al., 2019)	Reading Comprehension (RC)
Logical	**Conclusion:** One employee supervises another who gets more salary than himself. **Fact:** [DOC] David, Jack and Mark are colleagues in a company. David supervises Jack, and Jack supervises Mark. David gets more . . .	LogiQA (Liu et al., 2020)	Reading Comprehension (RC)
Multi-hop	**Question:** At which university does the biographer of John Clare teach English Literature? **Passage:** [DOC] John Clare : John Clare was an English poet . . . [DOC] CMS College Kottayam : The CMS College is one . . .	HotpotQA (Yang et al., 2018)	Reading Comprehension (RC)
Hybrid	**Question:** What was the percentage change in gaming between 2018 and 2019? **Context:** [TAB] Server products and cloud services | 32, 622 | 26, 129 . . . [DOC] Our commercial cloud revenue, which includes Office . . .	TAT-QA (Zhu et al., 2021)	Question Answering (QA)
Quantitative	**Hypothesis:** Teva earns $7 billion a year. **Premise:** After the deal closes, Teva will generate sales of about $7 billion a year, the company said.	EQUATE (Ravichander et al., 2019)	Natural Language Inference (NLI)

Table 1: The demonstration of five representative reasoning types. Listed are the types, the example questions, the representative dataset, and their corresponding tasks. [DOC] and [TAB] indicates the start of a passage and a semi-structured table respectively. Here we regard **Question**, **Conclusion** and **Hypothesis** as *sentence*, and **Passage**, **Fact**, **Context** and **Premise** as *natural context* in Figure 1.

Datasets : HotPotQA & LogiQA


```
{
    "id": 12,
    "url": "https://en.wikipedia.org/wiki?curid=12",
    "title": "Anarchism",
    "text": [["Anarchism"], ["Anarchism is a <a href=\"political%20p
    "charoffset": [[[[0, 9]]], [[[0, 9], [10, 12], [13, 14], [15, 48
}
```


HotPotQA

b

There is no doubt that minors should be prohibited from smoking. However, we cannot explicitly ban the use of automatic cigarette vending machines in order to prevent minors from smoking.This ban is just like setting up roadblocks on the road to prohibit driving without a license.These roadblocks naturally prohibit driving without a license, but also block more than 99% of licensed drivers.

In order to evaluate the above argument, which of the following questions is the most important?

A.Does the proportion of underage smokers in the total number of smokers exceed 1%?

B.How much inconvenience does the ban on the use of automatic vending machines bring to adult cigarette buyers?

C.Whether the proportion of unlicensed drivers in the total number of drivers really does not exceed 1%

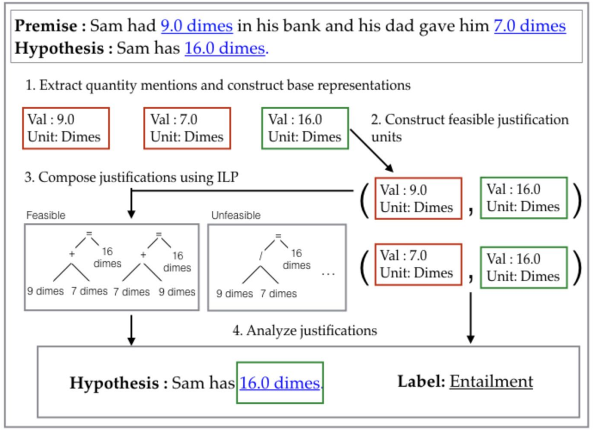
D.Is the harm of minor smoking really as serious as the public thinks?

LogiQA

Datasets : EQUATE & DROP

Q-Reas has five modules:

1. Quantity Segmenter: Extracts quantity mentions
2. Quantity Parser: Parses mentions into semantic representations called NUMSETS
3. Quantity Pruner: Identifies compatible NUMSET pairs
4. ILP Equation Generator: Composes compatible NUMSETS to form plausible equation trees
5. Global Reasoner: Constructs justifications for each quantity in the hypothesis, analyzes them to determine entailment labels



EQUATE

Leaderboard

DROP is a QA dataset which tests comprehensive understanding of paragraphs. In this crowdsourced, adversarially-created, 96k question-answering benchmark, a system must resolve multiple references in a question, map them onto a paragraph, and perform discrete operations over them (such as addition, counting, or sorting).

The leaderboard is powered by Beaker, AI2's powerful tool for rapid reproducible research.

Example DROP Question

Passage	Question	Answer
That year, his Untitled (1981), a painting of a haloed, black-headed man with a bright red skeletal body, depicted amid the artists signature scrawls, was sold by Robert Lehrman for $16.3 million, well above its $12 million high estimate.	How many more dollars was the Untitled (1981) painting sold for than the 12 million dollar estimation?	4300000
In 1517, the seventeen-year-old King sailed to Castile. There, his Flemish court . . . . In May 1518, Charles traveled to Barcelona in Aragon.	Where did Charles travel to first, Castile or Barcelona?	Castile
In 1970, to commemorate the 100th anniversary of the founding of Baldwin City, Baker University professor and playwright Don Mueller and Phyllis E. Braun, Business Manager, produced a musical play entitled The Ballad Of Black Jack to tell the story of the events that led up to the battle.	Who was the University professor that helped produce The Ballad Of Black Jack, Ivan Boyd or Don Mueller?	Don Mueller

DROP

TAT-QA

Revenue from external customers, classified by significant product and service offerings, was as follows:

(in **millions**)			
Year Ended June 30,	2019	2018	2017
Server products and cloud services	32,622	26,129	21,649
Office products and cloud services	31,769	28,316	25,573
Windows	20,395	19,518	18,593
Gaming	11,386	10,353	9,051
Search advertising	7,628	7,012	6,219
LinkedIn	6,754	5,259	2,271
Enterprise Services	6,124	5,846	5,542
Devices	6,095	5,134	5,062
Other	3,070	2,793	2,611
Total	$125,843	$110,360	$96,571

Our commercial cloud revenue, which includes Office 365 Commercial, Azure, the commercial portion of LinkedIn, Dynamics 365, and other commercial cloud properties, was $38.1 billion, $26.6 billion and $16.2 billion in fiscal years 2019, 2018, and 2017, respectively. These amounts are primarily included in Office products and cloud services, Server products and cloud services, and LinkedIn in the table above.

#	Reasoning	Question	Answer	Scale	Derivation
1	Word Matching (38.06%)	How much revenue came from Linkedin in 2018?	5,259	million	-
2	Set of spans (11.94%)	Which were the bottom 2 revenue items for 2017?	LinkedIn, Other	-	-
3	Comparison (5.65%)	Which year has the lowest revenue?	2017	-	-
4	Counting (2.28%)	How many revenue items are between 6,000 million and 6,500 million in 2019?	2	-	Devices ## Enterprise Services
5	Addition (2.37%)	What is the total revenue of commercial cloud from 2017 to 2018?	42.8	billion	26.6 + 16.2
6	Subtraction (16.17%)	How much of the total revenue in 2018 did not come from devices?	105,226	million	110,360 - 5,134
7	Division (3.84%)	How much does the commercial cloud revenue account for the total revenue in 2019?	30.28	%	38.1 billion / 125,843 million
8	Composition (19.69%)	What was the percentage change in gaming between 2018 and 2019?	9.98	%	(11,386 - 10,353) / 10,353

For more information, please read our ACL 2021 paper [PDF].

SVAMP (Simple Variations on Arithmetic Math word Problems)

PROBLEM: Text: Jack had 8 pens and Mary had 5 pens. Jack gave 3 pens to Mary. How many pens does Jack have now? Equation: 8 - 3 = 5
QUESTION SENSITIVITY VARIATION: Text: Jack had 8 pens and Mary had 5 pens. Jack gave 3 pens to Mary. How many pens does Mary have now? Equation: 5 + 3 = 8
REASONING ABILITY VARIATION: Text: Jack had 8 pens and Mary had 5 pens. Mary gave 3 pens to Jack. How many pens does Jack have now? Equation: 8 + 3 = 11
STRUCTURAL INVARIANCE VARIATION: Text: Jack gave 3 pens to Mary. If Jack had 8 pens and Mary had 5 pens initially, how many pens does Jack have now? Equation: 8 - 3 = 5

Table 1: Example of a Math Word Problem along with the types of variations that we make to create SVAMP.

POET - Program Executor

1. Numerical reasoning
2. Multi-hop reasoning
3. Logical reasoning

Comparison of Lines of Reasoning

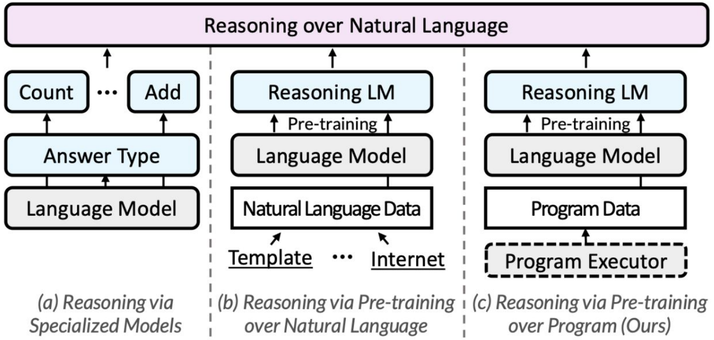


Figure 2: The illustration of different lines of reasoning, including **(a)** reasoning via specalized models, **(b)** reasoning via pre-training over natural language and **(c)** reasoning via pre-training over program (Ours).

Comparison: Programs vs Natural Language

Program

1. Logical form or Piece of code or Math expression

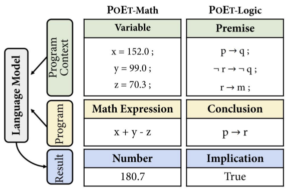
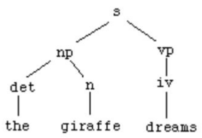


Figure 3: The illustration of POET-Math and POET-Logic. During pre-training, the concatenation of *program* and *program context* are fed into *language model* and the model is expected to output *result*.

2. Grammar Rules
3. Subset of English Vocabulary
   a. While, if-then-else, print constructs

Natural Language

1. Words, Sentences, Paragraphs
2. Grammar



3. Open class words vs closed class words

- Categories that will usually be open classes:
  - adjectives
  - adverbs
  - nouns
  - verbs (except auxiliary verbs)
  - interjections
- Categories that will usually be closed classes:
  - auxiliary verbs
  - clitics
  - coverbs
  - conjunctions
  - determiners (articles, quantifiers, demonstrative adjectives, and possessive adjectives)
  - particles
  - measure words or classifiers
  - adpositions (prepositions, postpositions, and circumpositions)
  - preverbs
  - pronouns
  - contractions
  - cardinal numbers

Comparison: Programs vs Natural Language

Program

1. Several programs can give same output
   a. While t < 6
      - print(t)
      - t +=1
   b. for(i=0;i++;i<6)
      - print(i)
2. **Same program** - can give **same output** for a **same input**
   a. While t < 6 and day == today:
      - t+= 1
   b. print(t)
3. **Expressive power** of Programs varies (Most expressive - Scala while least expressive - C) - **i < 5, i <= 4, i >= 4**
4. **Not ambiguous**

Natural Language

1. **Several sentences convey same meaning**
   a. Atmost 6 tsp of sugar a day
   b. No more than 6 tsp of sugar a day
2. **Same sentence** - can be **interpreted differently** in different contexts
   a. no more than 6 teaspoons per day is healthy
3. **High Expressive Power** (e.g generalized quantifiers)
   a. Fewer than 5, no more than 4, atleast 4
4. **Ambiguous**

Context

Program Context

- Most programming languages -> context-sensitive
- Variables serve as pivot points
- Connecting program context with program.

Math, logical forms : not context-free

SQL : Non-regular context free (Backus Naur form, normalization) and most expressive

- SELECT, FROM, WHERE, ORDER BY etc

Python, C, C++ are context-sensitive languages

- Python's indentation levels - suggest context-sensitiveness

Natural Language Context

- Context-sensitive
- sentence to natural context ⇔ program to program context

Natural language: context sensitive

English

Program Understanding

- Roberto Giacobazzi’s talk on A Complete Journey into (in)Completeness: Program Understanding - CSC Colloquium ->
- suggests program inputs and outputs are generally constant, it can be bounded, thus abstract interpretation is possible.
- Abstract interpretation - theory of soundness of semantics of computer programs.

POET-Math

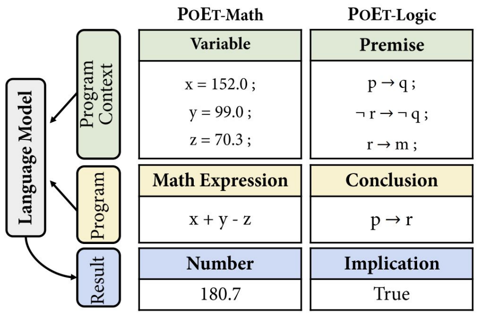


Figure 3: The illustration of POET-Math and POET-Logic. During pre-training, the concatenation of *program* and *program context* are fed into *language model* and the model is expected to output *result*.

- What is the difference in casualty numbers between Bavarian and Austrian?”
- **calculate the math expression**
- **Program:** math equation (+, - only)
- **Program Context:** values of each of variables
- **Result:** result of addition/subtraction
- Encoder-decoder only.

```python
import random
from random import shuffle
import os
from tqdm import tqdm


def expand_numbers_in_text(text, delim=" ", ignore_chars=[","], reverse_num=False):
    number_pattern = r"[-+]?[.]?[\d]+(,\d+)*[\.]?\d*(?:[eE][-+]?\d+)?%?"
    num_char_spans = [(m.start(0), m.end(0)) for m in re.finditer(number_pattern, text)]
    if len(num_char_spans) == 0: return text
    out_text = ""
    last_e = -1
    for i, (s, e) in enumerate(num_char_spans):
        out_text += text[:s] if i == 0 else text[last_e:s]
        num_str = delim.join([c for c in list(text[s:e]) if c not in ignore_chars])
        out_text += num_str if not reverse_num else num_str[::-1]
        last_e = e
    out_text += text[last_e:]  # append rest
    return out_text


def random_sample_numbers(with_vars):
    # the number of var_numbers
    op_num = random.randint(1, 2)
    candi_num = 30
    text_mapping = [chr(i) for i in list(range(65, 91)) + list(range(97, 122))]
    shuffle(text_mapping)
    var_numbers = []
    real_numbers = []
    candidate_numbers = []
    for i in range(candi_num):
        # random sample a number
        # 1000 float number
        is_int = random.randint(0, 9) < 8
        if is_int:
            final_num = str(random.randint(1, 100))
        else:
            final_num = str(random.randint(1, 1000) / 10)
        if i <= op_num:
            var_numbers.append(text_mapping[i])
            real_numbers.append(final_num)
            # random sample a + and -
            operator = random.choice(["*", "/"])
            if i != op_num:
                var_numbers.append(operator)
                real_numbers.append(operator)
        if i >= op_num and not with_vars:
            break
        candidate_numbers.append(final_num)
    if with_vars:
```

POET - Logic

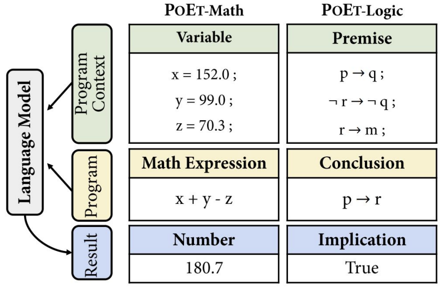


Figure 3: The illustration of POET-Math and POET-Logic. During pre-training, the concatenation of *program* and *program context* are fed into *language model* and the model is expected to output *result*.

- "Only if the government reinforces basic education can we improve our nation's education to a new stage. In order to stand out among other nations, we need to have a strong educational enterprise."
- **Program** - conclusion statement (first order logic)
- **Program Context** - Given a few first-order logic premise statements as the program context
- **Execution Result** - true or false (implication relationship between the program and the program context)
- Encoder Only
- Z3 SMT solver to generate synthetic data

185 lines (155 sloc) | 6.36 KB

```python
from z3 import *
import random
from random import shuffle
from itertools import combinations, product
from typing import List, Tuple
from functools import partial
from tqdm import tqdm
import os


solver = Solver()
vars_all_candidates = [chr(i) for i in list(range(97, 122))]
for symbol in vars_all_candidates:
    # init all variables
    exec("{0} = Bool('{0}')".format(symbol))


def sample_single_logic(var_inputs: Tuple):
    var_1, var_2 = var_inputs
    # given two vars, sample a logic to represent these
    # decide order of two vars
    logic_var_1 = var_1
    logic_var_2 = var_2
    if random.random() > 0.5:
        var_1 = "not {}".format(var_1)
        logic_var_1 = "Not({})".format(logic_var_1)

    if random.random() > 0.5:
        var_2 = "not {}".format(var_2)
        logic_var_2 = "Not({})".format(logic_var_2)

    # implies of two logic var
    if random.random() > 0.5:
        var_1, var_2 = var_2, var_1
        logic_var_1, logic_var_2 = logic_var_2, logic_var_1

    text = "( {} -> {} ) ;".format(var_1, var_2)
    logic = "Implies({}, {})".format(logic_var_1, logic_var_2)
    return logic, text


def sample_simple_hypo(var_candidates: List):
    # sample 1 or 2
    sample_num = 1 if random.random() < 0.75 else 2
    var_combinations = list(combinations(var_candidates, 2))
    shuffle(var_combinations)
    if sample_num == 1 or len(var_combinations) == 1:
        # select the first one as the var_candidates
        var_1, var_2 = var_combinations[0]
        return sample_single_logic((var_1, var_2))

    sample_predicate = "And" if random.random() < 0.75 else "Or"
    var_1, var_2 = var_combinations[0]
    logic_1 = sample_single_logic((var_1, var_2))
```

POET - SQL

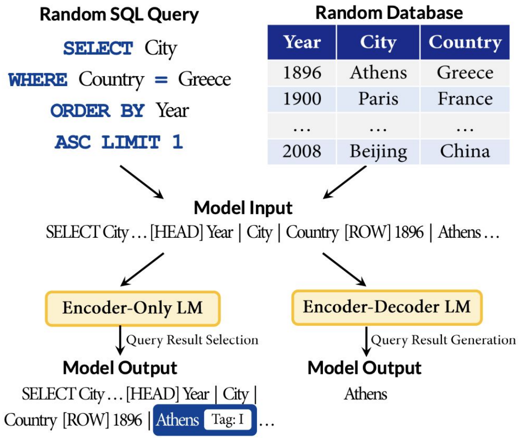


Figure 5: The illustration of POET-SQL pre-training tasks: query result selection for encoder-only and query result generation for encoder-decoder LMs.

Type	Example SQL Program
Arithmetic	SELECT [COL]$_1$ - [COL]$_2$
Superlative	SELECT MAX([COL]$_1$)
Comparative	SELECT [COL]$_1$ WHERE [COL]$_2$ > [VAL]$_2$
Aggregation	SELECT COUNT([COL]$_1$)
Union	SELECT [COL]$_1$ WHERE [COL]$_2$ = [VAL]$_2$ OR [COL]$_3$ = [VAL]$_3$
Nested	SELECT [COL]$_1$ WHERE [COL]$_2$ IN (SELECT [COL]$_2$ WHERE [COL]$_3$ = [VAL]$_3$)

Table 2: The six typical SQL programs that require reasoning. Listed are the type and the example SQL programs. [COL] and [VAL] represent the table column and the table cell value, respectively.

POET - SQL

1. SQL query,
2. a database, and
3. a query result
4. database is flattened into a sequence when it is fed into LMs
5. Encoder only LMs have insufficient expressiveness to produce out-of-context query results
6. query result generation -> uses table as Program Context

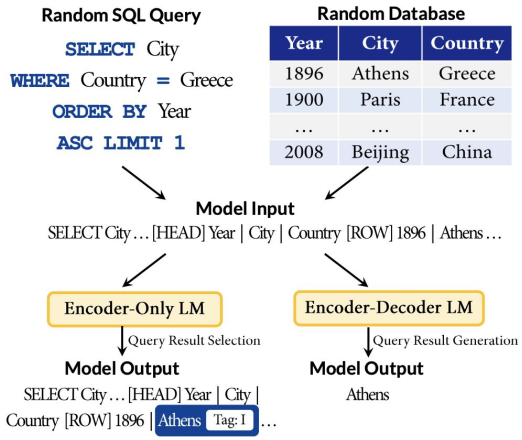


Figure 5: The illustration of POET-SQL pre-training tasks: query result selection for encoder-only and query result generation for encoder-decoder LMs.

SQUALL and WikiSQL (SQL to NL alignment datasets)


```
"nl": [
  "what",
  "is",
  "the",
  "difference",
  "in",
  "years",
  "between",
  "constiuency",
  "1",
  "and",
  "2",
  "?"
],
```

```
"nl_pos": [
  "WP",
  "VBD-AUX",
  "DT",
  "NN",
  "IN",
  "NNS",
  "IN",
  "NN",
  "CD",
  "CC",
  "CD",
  "."
],
```

```
"nl_ner": [
  "O",
  "O",
  "O",
  "O",
  "O",
  "DURATION",
  "O",
  "O",
  "NUMBER",
  "O",
  "NUMBER",
  "O"
],
```


SQUALL

Tables

These files are contained in the `*.tables.jsonl` files. A line looks like the following:


```
{
  "id":"1-1000181-1",
  "header":[
    "State/territory",
    "Text/background colour",
    "Format",
    "Current slogan",
    "Current series",
    "Notes"
  ],
  "types":[
    "text",
    "text",
    "text",
    "text",
    "text",
    "text"
  ],
  "rows":[
    [
      "Australian Capital Territory",
      "blue/white",
      "Yaa\u00b7nna",
      "ACT \u00b7 CELEBRATION OF A CENTURY 2013",
      "YIL\u00b700A",
      "Slogan screenprinted on plate"
    ],
    [
      "New South Wales",
      "black/yellow",
      "aa\u00b7nn\u00b7aa",
      "NEW SOUTH WALES",
      "BX\u00b799\u00b7HI",
      "No slogan on current series"
    ],
    [
      "New South Wales",
      "black/white",
      "aaa\u00b7nna",
```


Content and format

Inside the data folder you will find the files in `jsonl` and `db` format. The former can be read line by line, where each line is a serialized JSON object. The latter is a SQLite3 database.

Question, query and table ID

These files are contained in the `*.jsonl` files. A line looks like the following:


```
{
  "phase":1,
  "question":"who is the manufacturer for the order year 1998?",
  "sql":{
    "conds":[
      [
        0,
        0,
        "1998"
      ]
    ],
    "sel":1,
    "agg":0
  },
  "table_id":"1-10007452-3"
}
```


The fields represent the following:

- `phase`: the phase in which the dataset was collected. We collected WikiSQL in two phases.
- `question`: the natural language question written by the worker.
- `table_id`: the ID of the table to which this question is addressed.
- `sql`: the SQL query corresponding to the question. This has the following subfields:
  - `sel`: the numerical index of the column that is being selected. You can find the actual column from the table.
  - `agg`: the numerical index of the aggregation operator that is being used. You can find the actual operator from `Query.agg_ops` in `lib/query.py`.
  - `conds`: a list of triplets `(column_index, operator_index, condition)` where:
    - `column_index`: the numerical index of the condition column that is being used. You can find the actual column from the table.
    - `operator_index`: the numerical index of the condition operator that is being used. You can find the actual operator from `Query.cond_ops` in `lib/query.py`.
    - `condition`: the comparison value for the condition, in either `string` or `float` type.

WikiSQL

Data format

[sentence] col : [natural context]

POET-Roberta & POET-BART


```json
{
  "_name_or_path": "poet-roberta-large",
  "architectures": [
    "RobertaForMultiGoldSequenceLabeling"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "eos_token_id": 2,
  "gradient_checkpointing": false,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 1024,
  "initializer_range": 0.02,
  "intermediate_size": 4096,
  "layer_norm_eps": 1e-05,
  "max_position_embeddings": 1538,
  "model_type": "roberta",
  "num_attention_heads": 16,
  "num_hidden_layers": 24,
  "pad_token_id": 1,
  "position_embedding_type": "absolute",
  "transformers_version": "4.6.1",
  "type_vocab_size": 1,
  "use_cache": true,
  "vocab_size": 50265
}
```

Performance

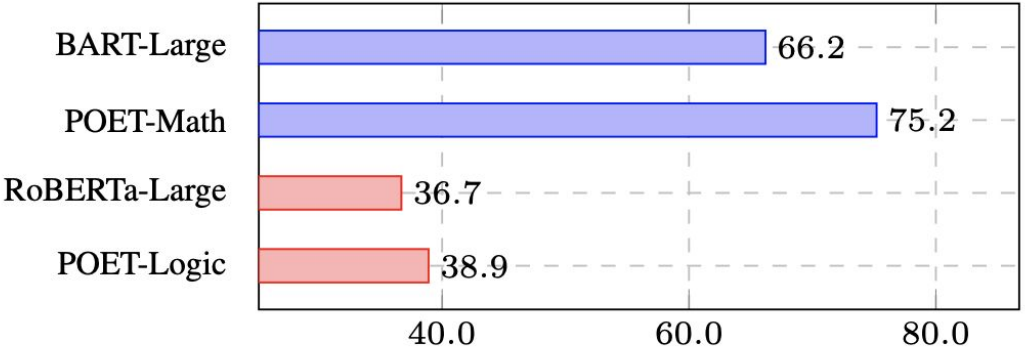


Figure 4: Fine-tuning EM performance [%] of different models on DROP (blue) and LogiQA (red).

Evaluation Claim 1

Will POET be affected by naturalness of program context or program? No.

1. Tuning the naturalness of program - we follow Liu et al. (2022) to **translate SQL queries into NL sentences** to make a more natural program, and **replace SQL reserved keywords with low-frequency tokens** to make a more **unnatural** program.
2. Tuning the naturalness of program context - POET-SQL **convert Database into a set of NL sentences**. Surprisingly,

Counter-evidence to the intuitive hypothesis: tuning the naturalness of program or program context do not significantly impact POET effectiveness.

Settings	EM	F_1
POET-SQL$_{BART}$	77.7	80.6
Tuning Program		
↪ *w.* Nnatural program	77.2	79.9
↪ *w.* Unnatural program	76.9	79.7
Tuning Program Context		
↪ *w.* Natural program context	76.5	79.0

Table 5: The EM and F_1 of POET-SQL$_{BART}$ on the DROP dev set with respect to different naturalness of program and program context.

Evaluation : Claim 2

Does pre-training on NL reasoning benefit model learning on program execution? Yes.

Test this - models pre-trained with NL reasoning would have better learnability on program execution.

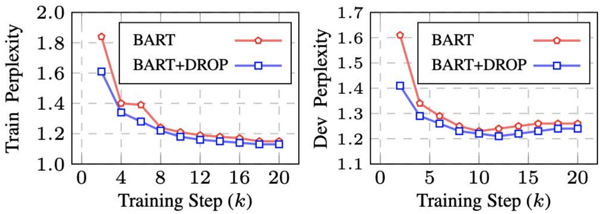


Figure 8: The train and dev perplexity of vanilla BART and BART pre-trained on DROP (BART+DROP) on the pre-training corpus of PoEt-SQL.

Evaluation : Claim 3

Can POET boost reasoning abilities of giant pre-trained language models? Yes.

apply POET-SQL to T5- 11B,. POET improves in boosting numerical reasoning abilities of giant LMs

Models	DROP♡		SVAMP
	EM	F1	EM
T5-11B	83.5	85.9	52.9
POET-SQL$_{T5}$	85.2 (+1.7)	87.6 (+1.7)	57.4 (+4.5)

Table 6: The experimental results of T5-11B and POET-SQL$_{T5}$ on test sets and dev sets (♡) of different datasets.

More questions

- Can POET now perform the NL tasks? Or has it “forgotten” previously learned information? How about unseen data?
- Can “context-free” language help to reason quantitative information in NL? It is not too clear.

Thank you!
Q&A