Code Generation for Data Wrangling Tasks

Sahil Bhatia

Data Wrangling

	A	B	C	D
1	Name	Price each month		
2		January	February	March
3	Entrenching Tool	$0.00	$32.00	$43.00
4	Biker Fuel Energy Bar	$0.00	$5.00	$5.00
5	Biker Fuel Energy Bar	$0.00	$12.00	$18.00
6	No-Hands Riding Kit	$250.00	$220.00	$180.00
7	Combination Lock	$30.00	$20.00	$15.00
8	Key Lo			
9	Stand			
10	Delux			
11	Execu			
12	Entre			
13	Biker			
14	Biker			
15	No-Ha			
16				

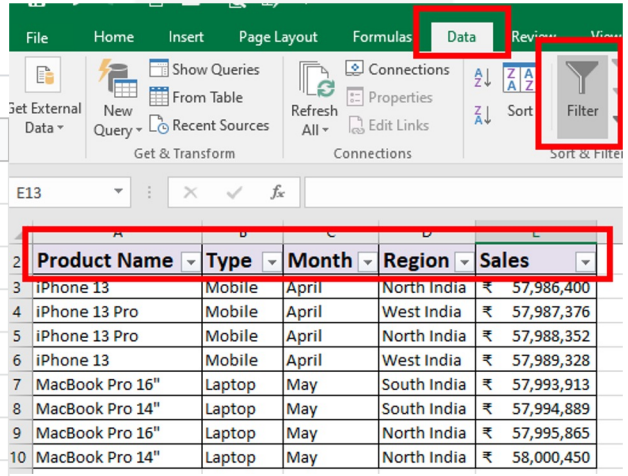
B2 fx =ArrayFormula(LEFT(A2:A7,6))

	A	B	C	D
1	**Phone number**			
2	+1 202-555-0158, US	+1 202		
3	+1 202-555-0153, US	+1 202		
4	+1 202-555-0112, US	+1 202		
5	+1 202-555-0164, US	+1 202		
6	+1 202-555-0132, US	+1 202		
7	+1 202-555-0121, US	+1 202		
8				
9				

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

df.fillna(0)

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	0.0
1	9	0.0	9.0	0	7.0
2	19	17.0	0.0	9	0.0



Data Wrangling

Data scientists spend **80%** of their **time** in **data wrangling**

Data Wrangling

Data Wrangling as PBE

```
Program := Concat(str,str) | Substring(str, id, id)
Str     := Dr. | F.N |  L.N
```

Examples

Synthesizer

Program

```
Concat(Dr., F.N)
```

F.N	L.N	Out
Sahil	Bhatia	Dr. Sahil
Abhishek	Shetty	Dr. Abhishek
Ajil	Jalal	Dr. Ajil

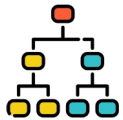
Data Wrangling as PBE

```
Program := Concat(str,str) | Substring(str, id, id)
Str     := Dr. | F.N |  L.N
```

Examples

F.N	L.N	Out
Sahil	Bhatia	Dr. Sahil
Abhishek	Shetty	Dr. Abhishek
Ajil	Jalal	Dr. Ajil

Synthesizer



Program

```
Concat(Dr., F.N)
```

Data Wrangling as PBE

Problem Statement

Can we use LLMs as synthesizer in PBE for data wrangling reliably?

Related Work

1. Can foundation models wrangle data?
    a. Not looking into DSL code generation
2. Several automated wrangling tools : FlashFill, Wrangler, FlashExtract

Data Wrangling as PBE

```
Program := Concat(str,str) | Substring(str, id, id)
Str     := Dr. | F.N |  L.N
```

Examples

F.N	L.N	Out
Sahil	Bhatia	Dr. Sahil
Abhishek	Shetty	Dr. Abhishek
Ajil	Jalal	Dr. Ajil

Synthesizer

Program

```
Concat(Dr., F.N)
```

1. No information about DSLs
2. No guarantee on correctness

Wrangling with LLMs (SyGuS)

```
(set-logic SLIA)

(synth-fun f ((name String)) String
  ((Start String (ntString))
   (ntString String (name " " "." "Dr."
            (str.++ ntString ntString)
            (str.replace ntString ntString ntString)
            (str.at ntString ntInt)
            (int.to.str ntInt)
            (str.substr ntString ntInt ntInt)))
   (ntInt Int (0 1 2
         (+ ntInt ntInt)
         (- ntInt ntInt)
         (str.len ntString)
         (str.to.int ntString)
         (str.indexof ntString ntString ntInt)))
   (ntBool Bool (true false
          (str.prefixof ntString ntString)
          (str.suffixof ntString ntString)
          (str.contains ntString ntString)))))


(declare-var name String)

(constraint (= (f "Nancy FreeHafer") "Dr. Nancy"))
(constraint (= (f "Andrew Cencici") "Dr. Andrew"))
(constraint (= (f "Jan Kotas") "Dr. Jan"))
(constraint (= (f "Mariya Sergienko") "Dr. Mariya"))
```

SyGuS LLM

```
The following program is written in SyGuS, which is an extension of SMT (Satisfiability Modulo Theories). Find a program f from
it's defined context-free grammar. The program should satisfy all the given constraints
{{Example1}}

{{Example2}}
```

Total Tasks : 108
Pass Rate : **34 (31%)**

SyGuS LLM (python)

```
(constraint (= (f "Nancy FreeHafer") "Dr. Nancy"))
(constraint (= (f "Andrew Cencici") "Dr. Andrew"))
```

```
assert f("Nancy FreeHafer") ==  "Dr. Nancy"
assert f( "Andrew Cencici") ==  "Dr. Andrew"
```

```
The goal is to synthesize a Python program f using only the string library functions that satisfy all the given assertions.
```

Total Tasks : 108
Pass Rate : **87 (80.5%)**

Model is good at solving task in python?

SyGuS LLM DSL(python)

Your task is to implement a Python Function `f` that satisfies the given assertions. You need to use only the set of provided functions and constants to achieve this. The following is an example of the function `f` and its assertions. Find the function `f` for the assertions in Example 2.

```
#Example1:
def concat(str1, str2):
return str1 + str2

def substring(str, start, end):
return str[start:end]

def int_to_str(int):
return str(int)

def str_to_int(str):
return int(str)

def index_of(str, sub, start):
return str.find(sub, start)

int_consts = [0,1,2,3,4,5,6,7,8,9,10]
str_const = ["Dr.", " "]

assert f("Nancy FreeHafer") == "Dr. Nancy"
assert f("Andrew Cencici") == "Dr. Andrew Cencici"
```

Total Tasks : 108
Pass Rate : **84 (78%)**

DSL code can be generated by simple syntax driven rules
Concat → str.++

SyGuS LLM DSL(python) - Failures

(constraint (= (f "+106 769-858-438") "+106 (769) 858-438"))

(constraint (= (f "+83 973-757-831") "+83 (973) 757-831"))

(constraint (= (f "+62 647-787-775") "+62 (647) 787-775"))

replace() takes 3 positional arguments but 4 were given

```
def f(phone):
    phone = replace(phone, "-", " (", 1)
    phone = replace(phone, "-", ") ", 1)
    return phone
```

SyGuS LLM DSL(python) - Failures

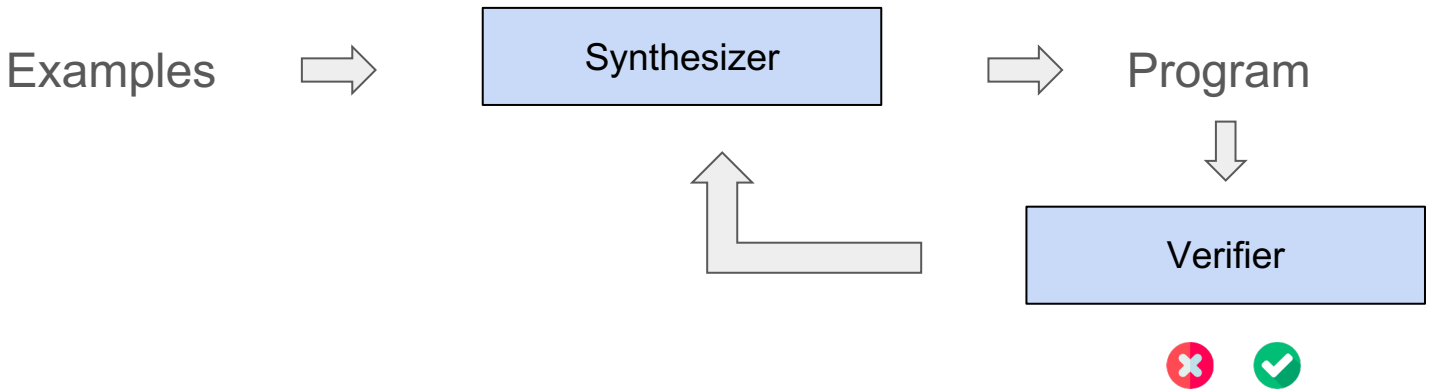
```
(constraint (= (f "+106 769-858-438") "106.769.858.438"))

(constraint (= (f "+83 973-757-831") "83.973.757.831"))
```

output 106769.858.438
excepted 106.769.858.438

```
def f(str):
    str = replace(str, " ", "")
    str = replace(str, "-", ".")
    return substring(str, 1, str_len(str))
```

Data Wrangling as PBE



Current Work - Feedback Loop with LLMs

1. Prompt in natural language if the solution is correct
    a. yes/no feedback
2. Use the feedback from the compiler/interpreter to fix the code
3. Use feedback from compiler + human feedback