Programming Aided by Machine Learning

Ollie Willette

Division of Science and Mathematics
University of Minnesota, Morris

April 22 2023

Introduction

Programming is hard.

- There are many places to make mistakes
- Switching between languages is especially difficult

In PHP
sizeof(myArray) ;
vs C++
myVector.size() ;

Figure – Me coding in APL (DALL-E 2)

Size of a Collection

Table – Retrieving Size of a Collection in Various Languages

Language	Unit	Method
Java	Standard array	.length
Java	String	.length()
Java	ArrayList	.size()
Clojure	Any collection	(count)
Python	Any collection	len()
Lua	Array	getn()
PHP	Array	sizeof()*
C++	Vector	.size()
C	Array	prayer

* sizeof() is a function in C, but does something different from what it does in PHP

Outline

Program Synthesis

What is program synthesis ?

- Complete program, all computer generated
- Not possible yet

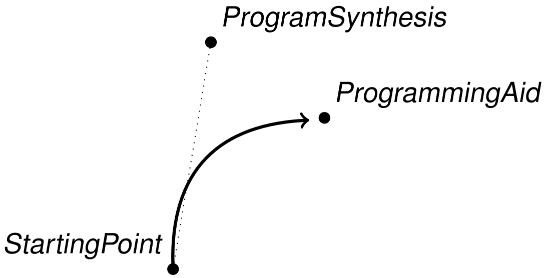
This talk is where program synthesis meets programming aid

Programming aid

- Anything that helps in programming
- Except program synthesis

Programming Aids

- Codex and AlphaCode were developed with program synthesis as the goal
- Did not meet the goal of program synthesis
- Both came closer than anything before them
- This progress can be used as a programming aid



Shoot for program synthesis, and if you fail you'll land among the programming aids

Large language models

GPT-3, Codex and AlphaCode are large language models (LLM)

Language model

- Predict the next word, like auto-complete on your phone
- Language models do this by using lots of math

Large ?

- They work on sequences of data
- These language models are called "large" because they use a lot of data
- Codex was trained using about 150 GB of Python code
- AlphaCode 715 GB of code

Codex (2021), AlphaCode (2022)

*LLM were invented by big chip makers to sell more storage and compute.

What are Transformers ?

Full explanation

What are Transformers ?

Full explanation Length : 1 day, 4 hours, 25 minutes

What are Transformers ?

Full explanation Length : 1 day, 4 hours, 25 minutes

Condensed explanation

What are Transformers ?

Full explanation	Length : 1 day, 4 hours, 25 minutes
Condensed explanation	Length : 1 hour, 22 minutes

What are Transformers ?

Full explanation Length : 1 day, 4 hours, 25 minutes

Condensed explanation Length : 1 hour, 22 minutes

Figure – Image by Amazon.com

- A type of neural network
- Self-attention to weigh the importance of different tokens
- Revolutionized natural language processing (NLP)
- The basis for LLM

Neural Network

How is a neural network built

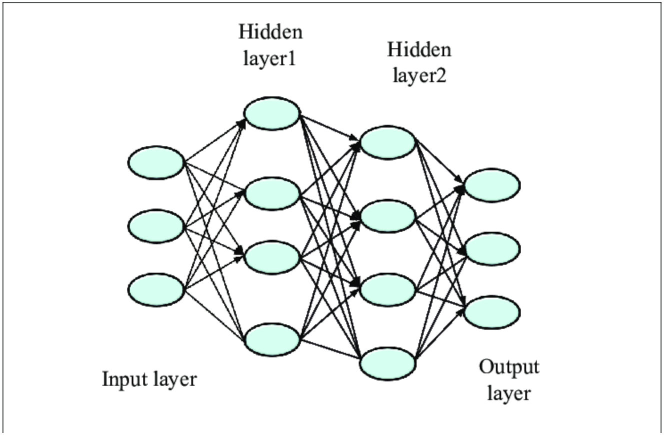


Figure – A diagram of a simple neural network

Image from www.researchgate.net

Attention

Attention ?

- Transformers are NN, but pay attention to their data
- This mean they have specific nodes dedicated to finding the most important parts of the input
- What the nodes pay attention to changes with every value

Encode/Decoder

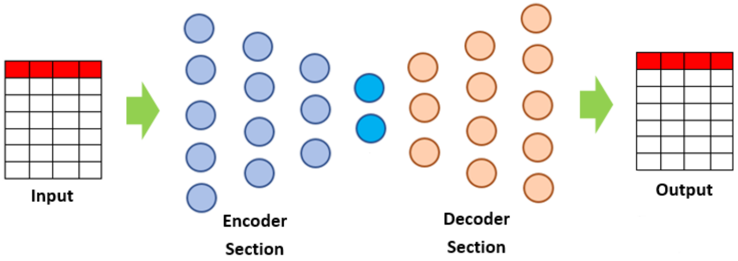


Figure – Encoder Decoder Architecture

Image from www.subscription.packtpub.com/

Codex

OpenAI used GPT-3 to develop Codex

- Codex writes code
- Specifically Python code, as it was trained only on Python code from GitHub
- Codex achieved a successful completion of 70.2% of problems given (easy)
- Codex solved 3.08% of competitive programming problems

Codex (2021)

Easy Programming Problem

An example of one of the problems Codex solved

```
def words_string(s):
"""
You will be given a string of words separated by
commas or spaces. Your task is to split the string
into words and return an array of the words. For
example:
words_string("Hi, my name is John") == ["Hi", "my",
"name", "is", "John"]
words_string("One, two, three, four, five, six") ==
["One", "two", "three", "four", "five", "six"]
"""
```

Codex (2021)

Simple Problem Result

Codex solved the problem with the below code

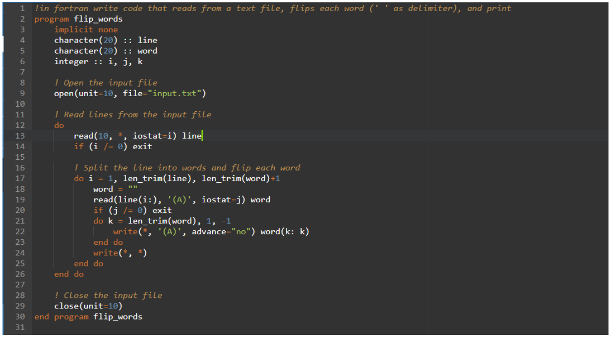
```
words = []
splitted_word = ""
for char in s:
    if char == " ":
        if splitted_word != "":
            words.append(splitted_word)
            splitted_word = ""
    elif char != ",":
        splitted_word += char
    if splitted_word != "":
        words.append(splitted_word)
return words
```

Codex (2021)

Codex Examples

This version of Codex is better than the one used in the paper, it is based on GPT-3.5, and is familiar with most programming languages
When given the prompt :

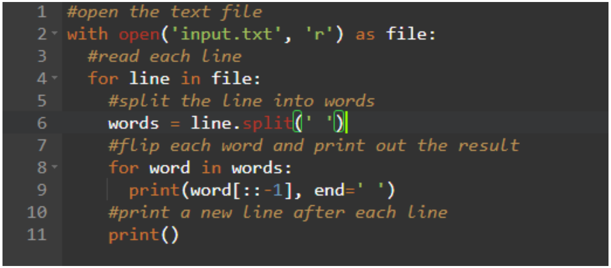
```
!in FORTRAN write code that reads from a  text file, flips
each word (’ ’ as delimiter), and print
```



```fortran
!in fortran write code that reads from a text file, flips each word (' ' as delimiter), and print
program flip_words
    implicit none
    character(20) :: line
    character(20) :: word
    integer :: i, j, k

    ! Open the input file
    open(unit=10, file="input.txt")

    ! Read lines from the input file
    do
        read(10, *, iostat=i) line
        if (i /= 0) exit

        ! Split the line into words and flip each word
        do i = 1, len_trim(line), len_trim(word)+1
            word = ""
            read(line(i:), '(A)', iostat=j) word
            if (j /= 0) exit
            do k = len_trim(word), 1, -1
                write(*, '(A)', advance="no") word(k: k)
            end do
            write(*, *)
        end do
    end do

    ! Close the input file
    close(unit=10)
end program flip_words
```

Python

When asking the same thing with Python



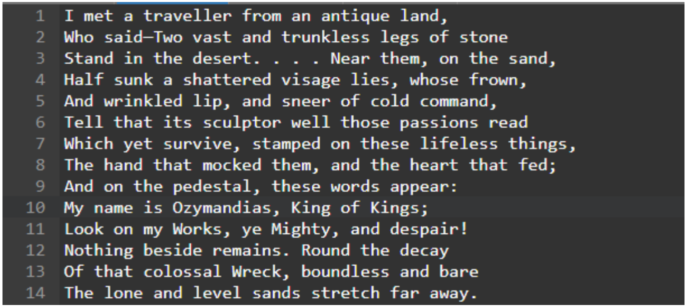
```python
#open the text file
with open('input.txt', 'r') as file:
  #read each line
  for line in file:
    #split the line into words
    words = line.split(' ')
    #flip each word and print out the result
    for word in words:
      print(word[::-1], end=' ')
    #print a new line after each line
    print()
```

Notes :

- Python had a much smaller program
- FORTRAN is also a much less common language on the internet

Ozymandias

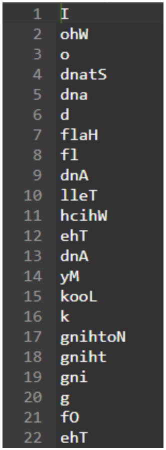
For test input I am using the poem "Ozymandias" by Percy Bysshe Shelley



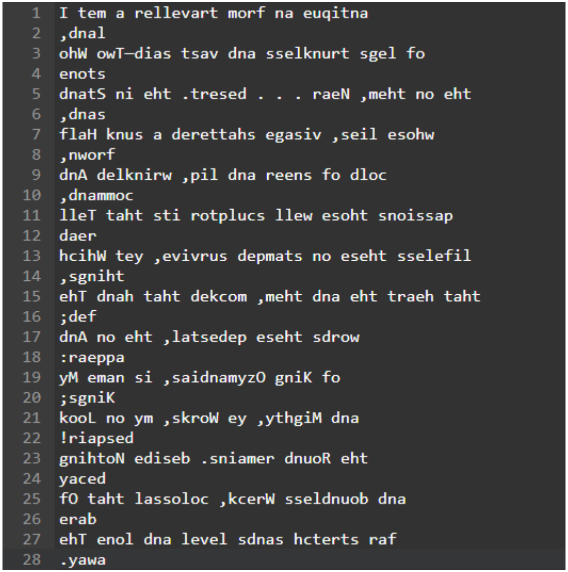
```
I met a traveller from an antique land,
Who said—Two vast and trunkless legs of stone
Stand in the desert. . . . Near them, on the sand,
Half sunk a shattered visage lies, whose frown,
And wrinkled lip, and sneer of cold command,
Tell that its sculptor well those passions read
Which yet survive, stamped on these lifeless things,
The hand that mocked them, and the heart that fed;
And on the pedestal, these words appear:
My name is Ozymandias, King of Kings;
Look on my Works, ye Mighty, and despair!
Nothing beside remains. Round the decay
Of that colossal Wreck, boundless and bare
The lone and level sands stretch far away.
```

Results

FORTRAN (below) vs. Python (right)

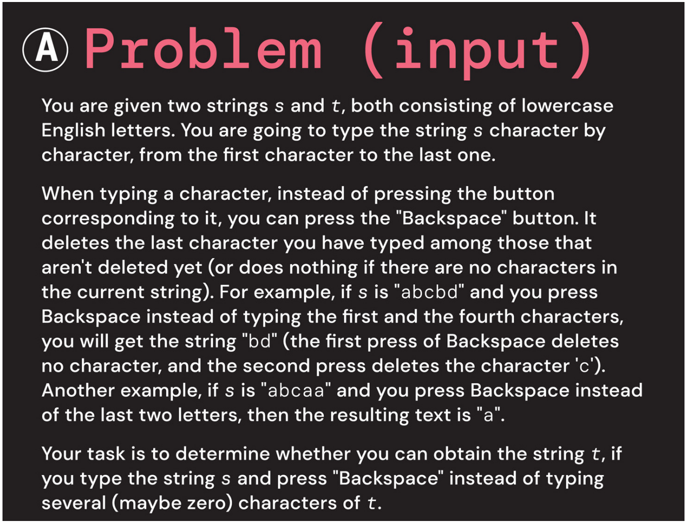


```
I
ohW
o
dnatS
dna
d
flaH
fl
dnA
lleT
hcihW
ehT
dnA
yM
kooL
k
gnihtoN
gniht
gni
g
fO
ehT
```



```
I tem a rellevart morf na euqitna
,dnal
ohW owT—dias tsav dna sselknurt sgel fo
enots
dnatS ni eht .tresed . . . raeN ,meht no eht
,dnas
flaH knus a derettahs egasiv ,seil esohw
,nworf
dnA delknirw ,pil dna reens fo dloc
,dnammoc
lleT taht sti rotpluc llew esoht snoissap
daer
hcihW tey ,evivrus depmats no eseht sselefil
,sgniht
ehT dnah taht dekcom ,meht dna eht traeh taht
;def
dnA no eht ,latsedep eseht sdrow
:raeppa
yM eman si ,saidnamyzO gniK fo
;sgniK
kooL no ym ,skroW ey ,ythgiM dna
!riapsed
gnihtoN ediseb .sniamer dnuoR eht
yaced
fO taht lassoloc ,kcerW sseldnuob dna
erab
ehT enol dna level sdnas hcterts raf
.yawa
```

Competitive Programming Problem


Ⓐ Problem (input)

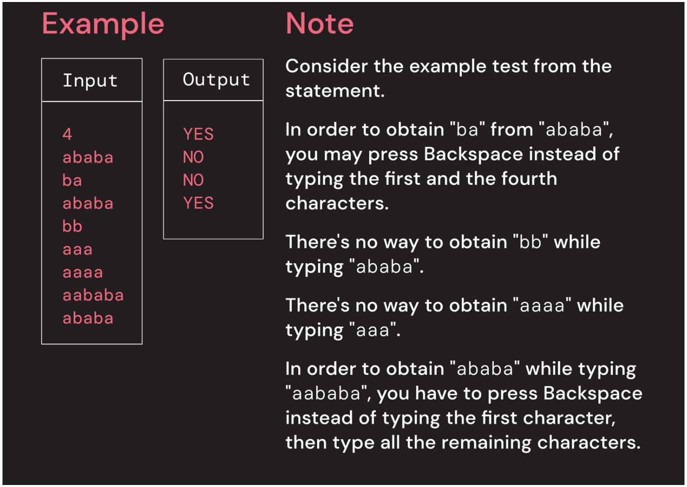
You are given two strings s and t, both consisting of lowercase English letters. You are going to type the string s character by character, from the first character to the last one.

When typing a character, instead of pressing the button corresponding to it, you can press the "Backspace" button. It deletes the last character you have typed among those that aren't deleted yet (or does nothing if there are no characters in the current string). For example, if s is "abcbd" and you press Backspace instead of typing the first and the fourth characters, you will get the string "bd" (the first press of Backspace deletes no character, and the second press deletes the character 'c'). Another example, if s is "abcaa" and you press Backspace instead of the last two letters, then the resulting text is "a".

Your task is to determine whether you can obtain the string t, if you type the string s and press "Backspace" instead of typing several (maybe zero) characters of t.

AlphaCode (2022)

Competitive Programming Problem 2nd half



Example

Input	Output
4 ababa ba ababa bb aaa aaaa aababa ababa	YES NO NO YES

Note

Consider the example test from the statement.

In order to obtain "ba" from "ababa", you may press Backspace instead of typing the first and the fourth characters.

There's no way to obtain "bb" while typing "ababa".

There's no way to obtain "aaaa" while typing "aaa".

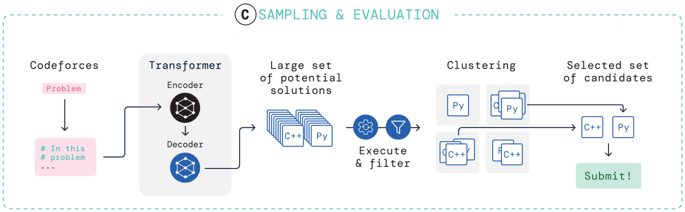
In order to obtain "ababa" while typing "aababa", you have to press Backspace instead of typing the first character, then type all the remaining characters.

AlphaCode (2022)

AlphaCode

Google built AlphaCode to solve competitive programming problems

- It had a solve rate of 29.6%
- Generated thousands of samples, and filtered them through the example given (99%)
- Had a second transformer that generated synthetic tests
- Samples would be clustered based on output to the synthetic tests



AlphaCode (2022)

Copilot Evaluation

GitHub released a tool called Copilot

- Programming aid based on Codex
- Gaining popularity among programmers
- Vaithilingam tested how useful Copilot was for programmers.

The study had 24 users

- 8 for 3 different programming tasks (easy, medium, hard)
- Varying levels of programming experience (1 user with less than 2 years of experience, 14 with 2-5 years, and 9 with 5+ years of experience)
- For each task, 4 users would attempt the task first with intellisense (standard auto-complete, most programmers are familiar with it) then with Copilot. The other 4 would attempt the task first with Copilot, then with intellisense.

Copilot Evaluation (2022)

Copilot Evaluation Data

This is their data

- DNF means the user was not done with the task after 25 minutes (in red)
- I highlighted the fastest time for each task in green

Table – Copilot vs Intellisense times

	Task 1 - Easy		Task 2 - Medium		Task 3 - Hard	
	Intellisense	Copilot	Intellisense	Copilot	Intellisense	Copilot
	9 : 35	1 : 46	7 : 48	12 : 53	13 : 41	11 : 08
	3 : 50	3 : 57	15 : 52	16 : 45	13 : 43	11 : 05
	4 : 49	4 : 55	16 : 28	7 : 26	22 : 42	4 : 04
	9 : 04	6 : 18	14 : 16	15 : 05	13 : 06	DNF
	5 : 18	1 : 18	7 : 35	13 : 24	23 : 13	19 : 54
	15 : 54	7 : 52	12 : 39	DNF	4 : 48	DNF
	5 : 27	3 : 12	10 : 47	6 : 02	DNF	DNF
	2 : 09	20 : 12	8 : 30	DNF	DNF	9 : 19
Average Time	7 : 01	6 : 11	11 : 44	11 : 56	13 : 36	11 : 06
Overall average time for all tasks combined					10 : 23	9 : 18

Issues With Copilot, Codex

Codex has issues.

- Generating non working code is annoying
- Furthermore it confuses new programmers
- Copilot, LLM, have issues with misalignment and hallucination
- Codex was trained on code from GitHub, which is known to have malignant code. This might make it dangerous for new programmers to use

Conclusion

Copilot and Codex are useful tools
New programmers shouldn't use them
They should have input for test-cases

References

References

1 Mark Chen et al. Evaluating large language models trained on code (Codex) (2021)

2 Yujia Li et al. Competition-level code generation with AlphaCode (2022)

3 Priyan Vaithilingam et al. Expectation vs. experience : Evaluating the usability of code generation tools powered by large language models (2022)