

## 目的

「AI Programmer」は日本語で書かれた問題文を理解し、その問題を実行するためのプログラムを自動的に作成してくれる。翻訳のように、プログラミングの自動生成も実用化レベルに高めていくことは、今後必要になっていくのではと感じた。そこで、「AI Programmer」が現在どのレベルのプログラミングが可能なのか、どのように命令文を作成すると、より精度の高いプログラムを生成してくれるのかなどを調べてみたいと思い研究した。

## AI\_Programmer

日本語でやりたいことを入力すると、任意のプログラミング言語で出力してくれるサービス。  
対応している言語はpython, C++, JavaScriptなどの主要なプログラミング言語だけではなく、SQLなどのデータ管理を主としたプログラミング言語にも使える。

言語を選択し、AI Programmerに書かせたいコードの処理内容を  
を入力してください。

C++

コード生成

グラフ

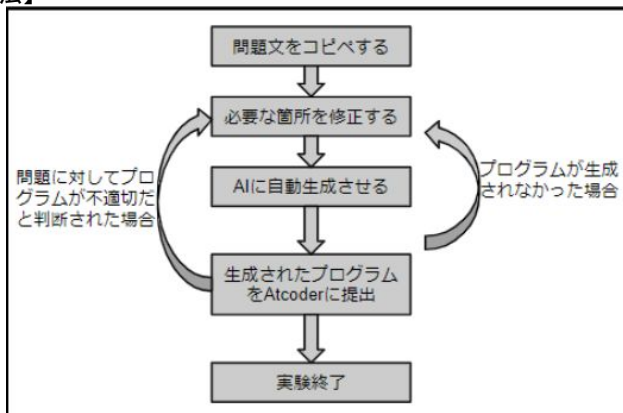
## 実験1 A問題の解法自動生成

A問題の例: 西暦 N 年は何世紀ですか？

### 【目的】

今回は、AI Programmerの200~210回目のabc問題の中でA問題についての実験を行う。A問題は生成するプログラムの内容自体は単純であるため、求められるのは高い精度の読解能力である。

### 【方法】



### 【結果】

A問題10/10 正解

### 【考察】

簡単に回答が出た。

## 実験2 B問題の解法自動生成

B問題の例: 0,1,6,8,9からなる文字列Sが与えられます。Sを180度回転したものを出力してください。

### 【実験内容】

B問題からは、ある程度のプログラミング能力や関数の使用が求められる場合があるため、プログラム知識が必要になってくる。

### 【方法】

実験の手順は実験1で説明された通りに行うが同時に現段階で分かっているAIの特性を反映した問題文の以下のような修正を行っていく。数字のみを出力する場合、「~の値を出力しなさい」と文末に付け加えるとよい/言葉で表すのが難しい計算手順などは数式を書くときよい/条件分岐はその条件ごとに文章を区切るとよい/理解するのが難しい説明は例を挙げるとよい

### 【結果】

B問題 9/10正解 プログラムがそこそこ長くなったものの、特に問題なく動いている。条件分岐の無駄や、全探索の取捨選択に問題がある。

### 【考察】

顕著な全探索のほうでは探索しなくてもよいところまで探索してしまうのが大きな問題点。これではC問題で通用しなくなってしまう。また、実験中に一度、出力されたプログラム中にコメントアウトが入っていた。全く指示していないにもかかわらず、なぜ必要のない文が含まれるのかがわからない。推測として、学習データで得られたプログラムの単語の出現確率で出力している、いわゆる言語モデルと同じと思われる。

## 実験3 C問題の解法自動生成

C問題の例: 数XをY回掛けたものを「XのY乗」といい、 $\text{pow}(X,Y)$ で表します。例えば  $\text{pow}(2,3)=2 \times 2 \times 2=8$  です。3つの整数A,B,Cが与えられるので、 $\text{pow}(A,C)$  と  $\text{pow}(B,C)$  の大小を比較してください。

### 【方法】

実験2と同様だが、同時に全探索などで無駄をなくすように誘導する分を追加する。

### 【結果】

C問題 1/10正解 ほぼ不正解となった。

### 【考察】

C問題ではほぼ同様に全探索や、再帰関数を用いる必要がある問題が多く見受けられ、事前の推察通り時間制限に間に合わずタイムアウトしてしまった。無駄をなくす誘導として時間制限をつける文言などを付け足すなどして対応を試みたが、うまくいかなかった。考えられる理由としては学習データの不足があげられる。1つの問題に対してC問題以上の難易度だと基本的に解答に類似点が少なく、解法が全く違うので今回対応できなかったと考える。

## 考察

### 【目的】

なぜ、C問題を解けなかったかのより深い考察と、逆にどのような問題に対して強いのかを考察したい。

### 【考察】

実験2で考察したように、このAIは言語モデルと同様の仕組みをしていると考える。また、このAI\_Programmerをリリースしている会社である有限会社ASRealはほかにも多種多様なAIを用いたサービスをリリースしており、中でもGPT-3という言語モデルを利用しているため、今回のAI\_ProgrammerもGPT-3を用いていると仮定する。この場合言語モデルとは、人間が話したり書いたりする言葉を単語の出現確率でモデル化したものであるため、必然的に出現確率の低いプログラムは出力されなくなってしまう。この仕組みでは、作者がモデルにABC問題を再学習させたとしても、そもそも全探索や、再帰関数を用いる問題の解答には類似点が少ないため、新しい問題が出てきたときに対応できない。逆に類似する問題が少ない、もしくは解答が類似する場合は多い問題には対応しやすい(例えばジュリア集合を描画するなど)。

## 今後の課題

今回行った実験ではC問題までのプログラムを解答させることは難しかった。今回の考察としてAI\_Programmerは協業プログラミングのような問題には相性が悪いという結論を出したが、実際にジュリア集合を描写させるなどの解答が少ないかつ、難しい問題を解けるのか確かめたい。