文字列に関する競技プログラミングにおける AIエージェントの性能向上

兼岩研究室　松平凌人

1.AIエージェント

AIエージェントとは、人の介入なしに、タスクを遂行するシステムのこと

AIエージェントと生成AIの違い
- 生成AI
  ユーザーの入力に対して、受動的に出力するだけ
- AIエージェント
  外部ツールや生成AIを使いながら、能動的にタスク実行を行う



AIエージェント

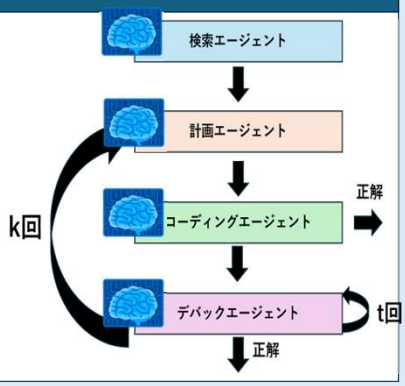
本研究では、競技プログラミングを解くAIエージェント「MapCoder」の改善を目指す

2.先行研究：MapCoder

競技プログラミングを解くAIエージェント

特色：
4つのエージェントによって構成されている

問題文を与えると、自動でコードを生成する



3.競技プログラミングを解く課題点

MapCoderでは**解けない競技プログラミングの問題**が多数ある

解くことができる問題例
　文字列の中にxが3つ以上続かないようにするために、削除するxの最小の数を求めよ

解くことができない問題例
　文字列の中にアルファベットが連続するものがあれば、文字を消すことができる。この時に、最大で消すことができる文字数を求めよ

生成されるチュートリアルの内容が不正確
検索エージェントにて生成されるチュートリアルが不正確だと、その先のエージェントに悪影響がでる

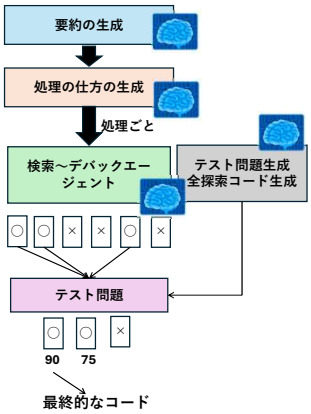
テスト問題の個数が少ない
不正解のコードもテストを通過してしまう

4.提案手法：AIエージェントの追加

課題点を解決するために、
新たなエージェントを追加

・チュートリアル生成エージェント
　チュートリアルを生成する前に、要約、問題の処理を生成させる
　この内容をもとにチュートリアルを生成させる

・テスト問題生成エージェント
　テスト問題(解答なし)を生成させる
　解答:全探索のコードを生成させ、そのコードが解いた解を解答とする



5.実験結果

文字列に関する問題1問を対象とした

表1 提案手法の正解率

実行回数	正解数	正答率(%)
5	5	5

AIエージェントは実行するたびに出力が変わる
表1より、実行を5回行って、すべて正解した
→対象としていた問題に対して適切なAIエージェントとなっている

表2 手法ごとの比較結果

手法	処理の仕方の個数	生成されたコード数	テスト問題の個数	2つのテストを合格したコード数
MapCoder	-	1	2	-
既存手法	**6**	**46**	**18**	**1**

処理の仕方を複数生成したため、生成されたコードの数が増えている
テスト問題の個数は、9倍となっている
2つのテストを合格したコードが1つのみ
→適切なテスト問題が生成されている

6.今後の課題

他の文字列に関する問題に対する汎用性の確認
データセットにある他の文字列に関する問題3問に対しての汎用性を確認

解答を自動で生成するエージェントの追加
全探索のコードは問題によっては実行時間が膨大になる
解答を正確に生成できるエージェントを作成し、追加する