BERT による競技プログラミングのジャンル推定

佐藤 利幸　　指導教員：小林 邦和

1　はじめに

競技プログラミングでは文章で書かれた問題を解くプログラムを作成することでプログラミングのスキルを競う．競技プログラミングの問題は数学の文章問題のような文章になっており，様々なシチュエーションの問題が揃っているため，これを人工知能が解くことができるのならば汎用的な問題解決ができる人工知能の開発に近づくことができる．競技プログラミングの問題の答えのプログラムを生成する手法は Y. Li らの研究 [1] がある．Y. Li らの提案手法では競技プログラミングサイト Codeforces[*1]で開催されたコンテストの中で 5000 人以上が参加したコンテストでは平均して上位 54.3% を達成している．この研究では Transformer[2] に問題文を入力してコードを生成しており，ジャンル推定は行っていない．競技プログラミングを解く際はまずどのようなアルゴリズムを使うかを考えるため，問題文のジャンルを正しく分類することが精度向上への道筋であると考えた．そこで，本研究では BERT[3] を用いて競技プログラミングの日本語の問題のジャンル推定を研究目的とする．

2　提案手法

BERT は，J. Devlin らによって提案された Transformer ベースの自然言語処理モデルである．発表された当時には多くの言語タスクにおいて他のモデルを上回る性能を示した．BERT の構造を図 1 に示す．

日本語の問題を対象とするため，日本の競技プログラミングサイト AtCoder[*2]の問題を対象とする．AtCoder の問題を構成要素に分けると，「実行時間制限」，「メモリ制限」，「問題文」，「制約」，「入出力の形式の指定」に分ける事ができる．本研究ではジャンル推定を対象としているため，BERT に入力するのは問題文と制約をペアにしたものとした．また，問題のラベル付けには AtCoder の問題のジャンル分類をしている AtCoder Tags[*3] を利用する．これは AtCoder ユーザーの投票により問題のジャンル分類を行う Web アプリである．

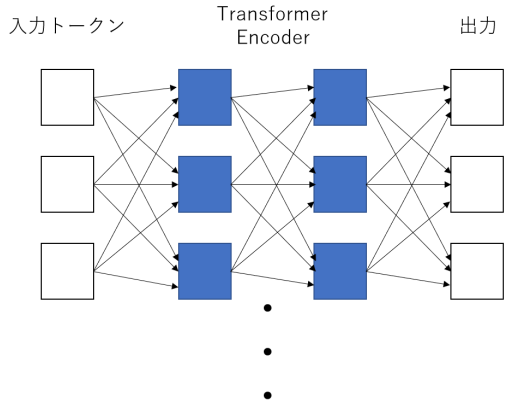

図 1　BERT の構造

3　計算機シミュレーション

本研究では BERT は東北大学が公開している事前学習済みモデル”bert-base-japanese-whole-word-masking”[4] を用いる．

AtCoder の問題を AtCoder Tags でのジャンル分類に沿ってデータセットを作成した．データ数は 4038 個となり，トークン化した際にトークン数が BERT の入力上限である 512 を超えるものは 317 個であった．これを訓練用データ，検証用データ，テスト用データで 6:2:2 で分割し提案手法を検証した．

表 1 に示す通り，テスト用データに対する正解率は 34.2% となった．また，難易度による正解率の違いを見るためテスト用データの中から AtCoder 上で最も頻繁に行われている”AtCoder Beginner Contest”の A 問題 (1 問目)，B 問題 (2 問目)，C 問題 (3 問目)，D 問題 (4 問目) に対する正解率を調べると，表 1 に示す通りとなった．基本的に A，B，C，D の順に難易度が高くなるように出題されている．このことから，人間にとって難易度の高い問題ほどジャンル推定も難しいことが確認できた．

表 1　提案手法の正解率

項目	正解率	データ数
テスト用データ全体	34.2%	818
A 問題	89.2%	37
B 問題	48.4%	62
C 問題	20.3%	59
D 問題	28.6%	56

4　おわりに

本研究では，BERT を用いることで競技プログラミングの問題のジャンル推定を行い，正解率 34.2% という結果を得た．しかし，学習初期で検証用データに対する損失が下がり切り以降損失が増加し続ける過学習が起こってしまった．今後の課題としては，過学習の原因がデータセットのデータ数が 4038 個と少ないことである可能性があるため，AtCoder 以外の問題を用いることによるデータセットの拡張が挙げられる．

参考文献

[1] Yujia Li, et al.: ”Competition-Level Code Generation with AlphaCode”, arXiv preprint arXiv:2203.07814 (2022)

[2] Ashish Vaswani, et al.:”Attention Is All You Need”,arXiv preprint arXiv:1706.03762 (2017)

[3] Jacob Devlin, et al.:”BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”,arXiv preprint arXiv:1810.04805 (2018)

[4] GitHub - cl-tohoku/bert-japanese at v1.0,GitHub,https://github.com/cl-tohoku/bert-japanese/tree/v1.0(2022/10/25 に参照)

[*1] https://codeforces.com/

[*2] https://atcoder.jp/

[*3] https://atcoder-tags.herokuapp.com/