プログラミング課題文からの重要箇所抽出

門井 仁弥 [1]　南條 浩輝 [2]　馬 青 [1]
[1] 龍谷大学理工学研究科　[2] 滋賀大学データサイエンス学部
[1]t22m006@mail.ryukoku.ac.jp　[2]hiroaki-nanjo@biwako.shiga-u.ac.jp　[1]qma@math.ryukoku.ac.jp

概要

本研究では BERT を用いてプログラミング課題文からの重要箇所の抽出を行った．具体的には，コード作成に必要な『入力』『出力』『条件』『繰り返し』が書かれた箇所を重要箇所とし，それらの抽出（ラベリング）を行う．複数の文が連なった課題文を対象とし，全体から重要箇所を直接ラベリングする方法と，段階的にラベリングする方法を提案する．後者は，第一段階で課題文を 1 文ずつ分類をし，第二段階でその分類情報を利用して系列ラベリングする方法である．直接ラベリングでは，出現頻度が低いラベルに対する精度（F(0.5)）が低いことがわかった．次に，段階的ラベリングを行ったところ，直接ラベリングよりも全てのラベルで F(0.5) の改善が得られた．特に，出現頻度が低いラベルに対して大きな改善が得られることがわかった．

1　はじめに

第 4 次産業革命と言われる IT 技術の進歩により，社会のデジタル化が進んでいる．情報技術の理解を深める観点から情報教育が推進されており，2020 年から小学校教育において，プログラミング教育が導入され，昨今，プログラミングはより身近な技能へとなってきている [1][2]．プログラミング教育の目的は，問題解決能力の育成であり，プログラミングを学ぶことで，論理的思考力や批判的思考力が鍛えられ，問題に対して，効果的な解決策を導き出す能力が向上することが期待されている．そのため，研究においても，プログラミング教育やそれを支援する研究が盛んになっている [3][4][5]．これらの研究の詳細は 2 節で述べる．

しかし，プログラミングのコードを作成する際，文章内から必要な情報を見つけ出し，手続きを組み立てて習得することが重要であるにも関わらず，このような研究は十分になされているとは言えない．そこで，われわれはコードを作成する際の必要な情報を見つけ出す支援に関する研究を行った．

2　課題と関連研究

2.1　プログラム支援の関連研究とその課題

人が課題文からソースコードを作成するには，図 1 のようなプロセスを踏んでいると考える [3]．

この考えから，プログラムのコードからプログラム手続きを生成する研究 [4] や，問題文からプログラム手続きを生成する研究もある [5]．しかし，これらの研究では，問題文内のどの語句が，必要な情報なのかが明確に提示されていない．

そこで本研究では，与えらえたプログラミング課題文に対して，プログラムコードを作成するのに必要な情報の可視化に取り組む．可視化ができれば，学習者が重要な情報を容易に見つけ出すことができるためである．可視化するためには，文中のどの箇所が必要な情報かを見つける必要がある．すなわち，コード作成にとって重要な情報を文中から抽出する必要がある．

2.2　情報抽出の関連研究

文から情報を抽出する研究は多くあり，障害レポートから情報抽出をおこなっている研究では，LSTM を用いて系列ラベリングとして解く手法 [6] や，BERT や T5 の質疑応答を応用して解く手法があ

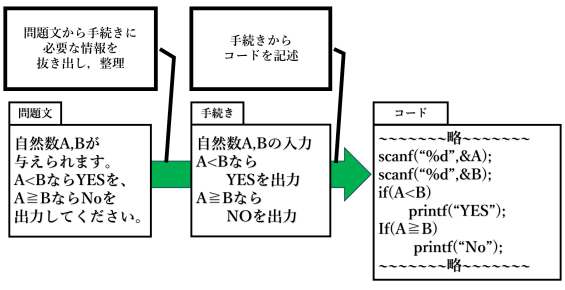

図 1　課題文からコードを記述する概要

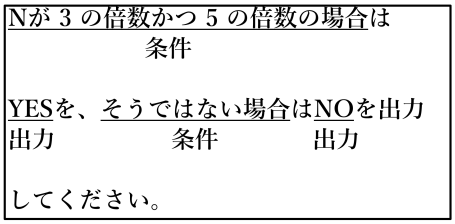

図 2　課題文の重要箇所の例

る [7]．他では，論文や判決書から情報抽出の研究がある [8][9]．しかし，プログラミング課題文から重要情報を抽出する研究は十分になされているとは言えない．

3　重要箇所の抽出

3.1　重要ラベルの定義

課題文中にある手続きを記述する際に，その根拠となる情報を重要情報と捉えて，重要情報がある文中の箇所を重要箇所とした．つまり，この重要箇所が可視化したい箇所であり，情報抽出したい箇所でもある．重要情報のラベルは，『入力』『出力』『条件』『繰り返し』の 4 つに設定した．

例えば，図 2 のような課題文があった場合，文中の「N が 3 の倍数かつ〜場合」は，ラベル『条件』の重要箇所であり，「YES」がラベル『出力』の重要箇所となる．

3.2　直接ラベリング

重要箇所の抽出には，BERT の系列ラベリングを用いた．課題文を単語系列として扱い，各単語に，設定した重要情報のラベルを付与する系列ラベリング問題として表現した．

図 3 のような例だと，対応する単語に『条件』と『出力』の付与されている

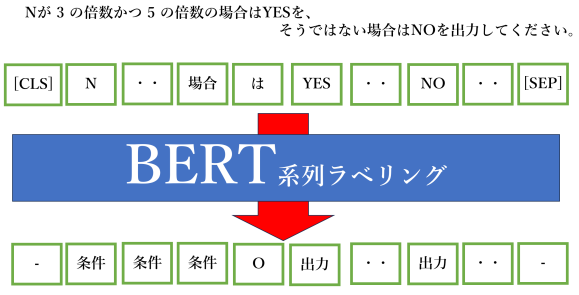

図 3　直接ラベリング

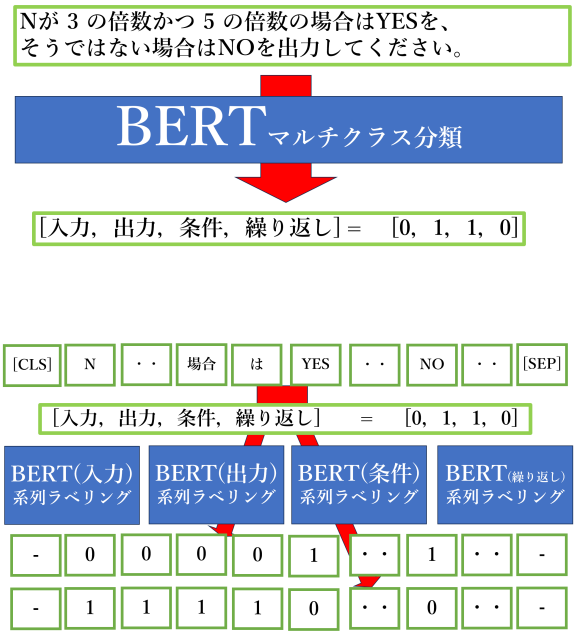

図 4　段階的ラベリング（上: 一段階目，下: 二段階目）

3.3　段階的ラベリング

直接ラベリングでは全てのラベルを同時に扱うため，出現頻度の低いラベルに対するラベリング精度が低いことが懸念される．そこで，まず課題文の文ごとに各ラベルが存在するかを分類し，次に存在するラベル専用の系列ラベリングモデルで箇所を特定するという段階的ラベリングを提案する．概要を図 4 に示す．それぞれの段階について，詳細に説明する．

3.3.1　一段階目のマルチクラス分類

概要は図 4 の上段に示されている．課題文に対して文ごとにどの重要情報（ラベル）が含まれているかを BERT のマルチクラス分類とする．各文に対してそれぞれの重要情報のラベルが存在するか真または偽（1 または 0）を同時に予測する．図 4 上段の例では，与えられた文に対し，『出力』と『条件』があると予測している．

3.3.2　二段階目の系列ラベリング

概要は図 4 の下段に示されている．二段階目は直接ラベリングと同様に BERT を用いた系列ラベリング問題とする．直接ラベリングと異なる点は，ラベルごとに，本研究では合計 4 つの BERT を用意する点である．各 BERT はそれぞれのラベルのみを系列ラベリングできるよう学習する．

ここでは，与えられた文に対して，一段階目で存在すると予測されたラベルのみを，各ラベル専用の BERT で系列ラベリングして求める．図 4 の下段では，前段で存在すると判定された『出力』と『条件』のそれぞれの重要箇所抽出する BERT を 2 つ用いる．それぞれの BERT の出力は，各単語が該当する重要箇所かを示す真か偽（1 か 0）の値である．

4 実験

4.1 コーパス

4.1.1 取得したプログラミング課題文

本研究ではオンラインプラットフォームである AtCoder と Paiza ラーニングの 2 つから提供されているプログラミング課題文を取得した．AtCoder[1] は，主に日本のプログラマ向けに設計されたオンライン競技プログラミングプラットフォームである．参加者は，様々なコーディングの問題を解くことができる．Paiza ラーニング[2] は，プログラミングスキルの学習と向上を目的としたオンライン学習サービスの一つである．日本のユーザーを対象に，初心者から経験者までさまざまなレベルの人が利用してる．

これらの取得した課題文に対してアノテーション（重要箇所へのラベル付与）を行い，コーパスを作成した．

4.1.2 重要箇所へのラベル付与と前処理

取得した課題文に対して，本研究で定義した重要箇所を表すラベルを人手で付与した．

また，インデントに使われている全角スペースを全て削除し，文分割をおこなった．このようにして，コーパスを作成した．

表 1 コーパスの詳細

	課題の文数	付与した箇所
入力	3377	4188
出力	2400	2652
条件	1002	1361
繰り返し	154	201
なし	5313	-

表 2 テストデータの詳細

	課題の文数	付与した箇所
入力	340	722
出力	204	588
条件	186	255
繰り返し	18	54
なし	480	-

1） https://atcoder.jp/?lang=ja
2） https://paiza.jp/works

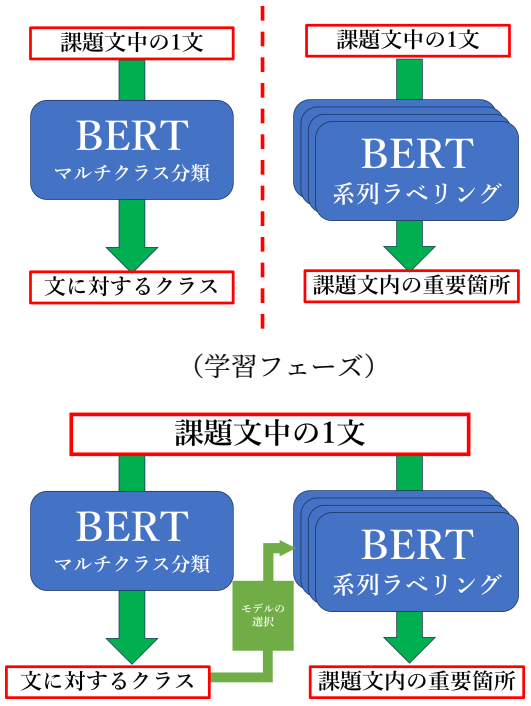

図 5 段階的ラベリングの学習 (上) と推論 (下) の概要

4.1.3 コーパスの詳細

本研究で作成したコーパスは，全 2,172 件の課題文からなる．課題文を 1 文に文分割しており，文の総数は 12,471 文である．コーパスの詳細を表 1 に示す．

2,172 件の課題文コーパスを学習データ 2,000 件とテストデータ 172 件に分割した．テストデータの詳細を表 2 に示す．1 文中に付与した箇所が複数ある場合があるため，付与した箇所は多くなっている．また 1 文中に設定した 4 つのラベルがない場合もある．このテストデータを用いて，評価をおこなう．

4.2 実験設定

学習方法において，直接ラベリングの場合は，モデルは 1 つで，入力は文全体である．段階的ラベリングでは，図 5 上段のように，一段階目と二段階目のモデルは独立で学習をおこない，二段階目の各モデルの訓練データには，必ず文中に該当する単語があるものを使用している．

実験には，BERT の事前学習済みモデルを用い，それぞれの本抽出タスクに対して fine-tuning した上で

表 3 直接ラベリングの評価

	Pre.	Rec.	F(0.5)
入力	0.749	0.842	0.792
出力	0.807	0.817	0.811
条件	0.228	0.687	0.338
繰り返し	0.307	0.680	0.391
ALL	0.657	0.815	0.777

表 4 段階的ラベリングの評価

	Pre.	Rec.	F(0.5)
入力	0.726	0.876	0.841
出力	0.874	0.850	0.861
条件	0.727	0.737	0.729
繰り返し	0.711	0.892	0.848
ALL	0.752	0.849	0.827

実施する．BERT のモデルとして，東北大学乾・鈴木研究室の Wikipedia で訓練済み日本語 BERT モデル (cl-tohoku/bert-base-japanese-whole-word-masking) [10] を使用した．

テストデータを除いたデータを 10 分割し，その内の 1 つを検証データとして，残り 9 つを訓練データとして交差検証をおこなうことでハイパーパラメータを設定した．

ハイパーパラメータについては，マルチクラス分類の BERT において optimizer は AdamW，学習率 1e − 5，バッチサイズ 4，エポック数は 6，ドロップアウト 0.1 に設定し学習した．また，系列ラベリングの BERT において optimizer は AdamW，学習率 1e − 5，バッチサイズ 8，エポック数は 3，ドロップアウト 0.2 に設定し学習した．

4.3 評価方法

評価指標として適合率 (Precision)，再現率 (Recall)，F 値を用いる．F 値は，式 (1) で与えられる．

$$F(\beta) = \frac{(\beta^2 + 1.0) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \quad (1)$$

本研究では，最終的にモデルから提示される重要箇所が，どれだけ正確に予測できているか (適合率が高い) を重視するため，F(0.5) で評価している．

ハイパーパラメータの選定の際にも，この F(0.5) が高いもので決定している．また，段階的ラベリングの二段階目のモデルの評価については，図 5 の下段のように一段階目で分類されたラベルからモデルからを選定し，重要箇所を抽出する手法になっているため，最終的に二段階目で出力された結果から評価している．

表 5 段階的ラベリングにおける第一段階（マルチクラス分類）の評価

	Pre.	Rec.	F(0.5)
入力	0.871	0.855	0.863
出力	0.953	0.927	0.940
条件	0.729	0.624	0.673
繰り返し	0.623	0.919	0.747
ALL	0.869	0.841	0.855

5 実験結果

直接ラベリングによる重要箇所抽出の結果を表 3 に，段階的ラベリングによる重要箇所抽出結果を表 4 に示す．ALL はマイクロ平均を表す．『入力』『出力』については F(0.5) の値が直接ラベリングでそれぞれ 0.792, 0.811，段階的ラベリングで 0.841, 0.861 と高い結果が得られている．『条件』『繰り返し』といった出現頻度が低いラベルに対しては，直接ラベリングでそれぞれ 0.338, 0.391 と精度が低かったのに対し，段階的ラベリングでは，それぞれ 0.729, 0.848 と，『入力』『出力』と同程度に高精度な抽出ができるようになったことがわかる．ほぼ全てにおいて，直接ラベリングよりも段階的ラベリングのほうが精度が高く，特に F(0.5) については全て高かった．段階的ラベリングにより，プログラミング課題文中の重要箇所の抽出が高精度にできることを示した．

最後に，段階的ラベリングの一段階目の各文に対するラベルが存在するかの検出結果（マルチクラス分類結果）を表 5 に示す．一段階目で『条件』『繰り返し』の抽出精度は『入力』『出力』よりも低いことがわかった．今後はこれらを向上させることで二段階目の性能向上を目指す予定である．

6 おわりに

本研究では，BERT でプログラミング課題文から重要箇所を抽出する手法を提案した．直接ラベリングと段階的ラベリングを提案し，段階的ラベリングが優れることを確認した．特に，出現頻度が低いラベルに対して大きな改善が得られることがわかった．今後は，段階的ラベリングの第一段階の文分類において，前後文の文脈情報を取り入れるなど改善を行いさらなる性能向上を図りたい．

謝辞

本研究は JSPS 科研費 19K12241 の助成を受けたものです.

参考文献

[1] 文部科学省. 教育の情報化の推進, 2018.Accessed December.22,2023. https://www.mext.go.jp/a_menu/shotou/zyouhou/index.htm.

[2] 文部科学省. プログラミング教育, 2018.Accessed December.22,2023. https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1375607.htm.

[3] Junko Shinkai, Yoshikazu Hayase, and Isao Miyaji. A trial of algorithm education emphasizing manual procedures. In **Society for Information Technology & Teacher Education International Conference**, pp. 113–118. Association for the Advancement of Computing in Education (AACE), 2016.

[4] Hiromitsu Shiina, Sakuei Onishi, Akiyoshi Takahashi, and Nobuyuki Kobayashi. Automatic comment generation for source code using external information by neural networks for computational thinking. **International Journal of Smart Computing and Artificial Intelligence**, Vol. 4, No. 2, pp. 39–61, 2020.

[5] 大西朔永, 椎名広光. Seq2seq の組み合わせによる問題文からの段階的プログラムコメント生成. 言語処理学会 第 27 回年次大会, pp. 1857–1861, 2021.

[6] 小平知範, 宮崎亮輔, 小町守. 障害情報レポートに対する同時関連文章圧縮. 言語処理学会第 23 回年次大会, p. 189–193, 2017.

[7] 山下郁海, 岡照晃, 小町守, 真鍋章, 谷本恒野. 日本語 T5 モデルを用いた障害レポートからの重要箇所抽出. 言語処理学会第 28 回年次大会, p. 986–991, 2022.

[8] 村田真樹, Stijn De Saeger, 橋本力, 風間淳一, 山田一郎, 黒田航, 馬青, 相澤彰子, 鳥澤健太郎. 論文データからの重要情報の抽出と可視化. 人工知能学会全国大会論文集, Vol. JSAI2009, pp. 3F2NFC39–3F2NFC39, 2009.

[9] 阪野慎司, 松原茂樹, 吉川正俊. 機械学習に基づく判決文の重要箇所特定. 言語処理学会第 12 回年次大会発表論文集, pp. 1075–1078, 2006.

[10] 東北大学乾・鈴木研究室 BERT モデル. cl-tohoku/bert-base-japanese-v2. https://huggingface.co/cl-tohoku/bert-base-japanese-v2.