

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

大西 朔永

岡山理科大学大学院総合情報研究科
i20im01os@ous.jp

椎名 広光

岡山理科大学総合情報学部
shiina@mis.ous.ac.jp

1 はじめに

第4次産業革命の中において、その核となる情報技術の理解を深める観点から、2020年より、プログラミング教育 [1, 2] が小学校において、導入されており、情報技術に直接関わりのない大学においても導入されている。また、エラーを可視化する学習支援ツールの研究 [3] や操作ログを用いた初心者向け教育の研究 [4, 5], カードの操作を基本とした学習システムの研究 [6] が行われている。プログラミング教育の主目的はプログラムの作成ではなく、論理的思考力の育成である。論理的思考力とは、課題を解決するために必要な処理内容と処理の順序を考える能力で、実際にプログラミングをせずとも、論理的思考力を習得することは可能であり、その方法として、手続き学習を提案している。

手続き学習の研究として、手作業によるアルゴリズムの学習についての研究 [7] がある。手続き学習には、プログラムの概要を表す問題文と手続きが必要となるが、人手によるプログラム手続きの作成は、コストが高く、困難である。そのため、Seq2Seq(Encoder-Decoder)[8, 9, 10] を用いたプログラムのソースコードからプログラム手続きを自動生成する手法 [11, 12, 13] の開発を行っている。また、Java のソースコードからメソッドのコメントを生成する研究 [14, 15, 16] も行われている。これらの手法ではプログラム手続きの生成にプログラムのソースコードが必要となる。

本研究では、図1に示すように、ソースコードを必要としないプログラム手続きの生成を目的に、問題文から直接プログラム手続きの生成を試みている。しかし、問題文から直接プログラム手続きにあたる行コメントを生成することは難しいと考えている。プログラムを作成する際の人間の思考過程は、最初に問題文から大まかな処理の流れ(手順)を考

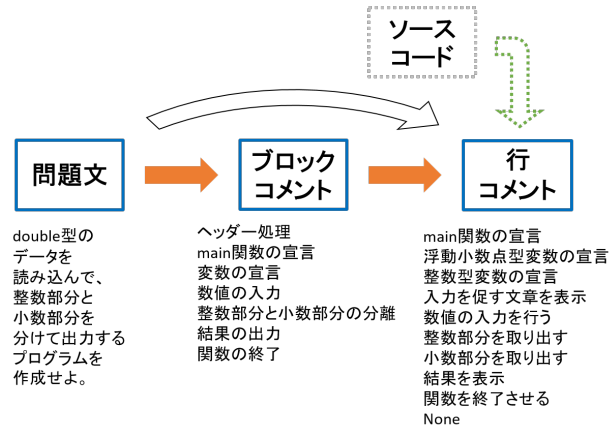


図1 問題文からの段階的プログラムコメント生成の概要

え、次に手順をより詳細な行単位の処理に具体化するという2段階であると考えられる。そのため、本研究では、問題文と行コメントの中間に大まかな処理の流れであるブロックコメントを加えた手法を提案している。GRU[17]を用いたSeq2Seqを組み合わせさせたモデルを用いて、最初に問題文からブロックコメントを生成し、次に生成されたブロックコメントから詳細な行コメントを生成する2段階の生成手法である。問題文から直接行コメントを生成する手法と人間の思考過程に則った2段階生成の手法を自動機械翻訳評価指標であるBLEU[18]やコメントの生成例で比較している。

2 手続き学習システム

プログラムには、プログラムの目的を達成するために必要な手続きとその手続きの順序を正しく記述する必要がある。必要な手続きをプログラムが動作する順序に並び替える問題を解くことにより、プログラムの手順を理解することが手続き学習の目的である。手続き学習システム [19] の流れは、最初にプログラムの問題文と必要な手続きを表示し、次にユーザに手続きをプログラムが動作する順序に並び替えて貰い、最後に、手続きの順序の正解、不正解

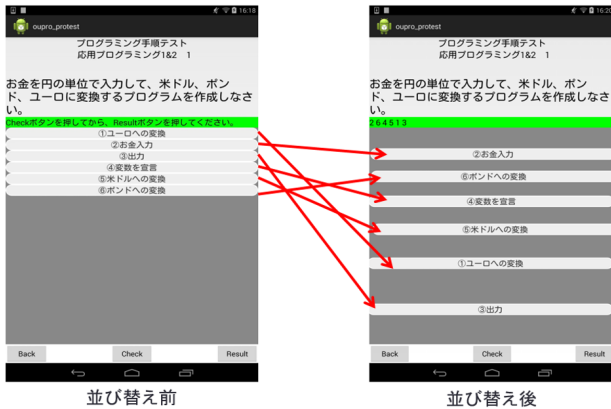


図2 手続き学習システム

を表示する。

手続き学習システムを図2に示す。例の為替のプログラムの場合、問題文は「お金を円の単位で入力して、米ドル、ポンド、ユーロに変換するプログラムを作成しなさい。」であり、そのプログラムに必要な手続き6個が表示されている。左の画面が並び替える前の問題画面であり、右の画面はユーザが並び替えた後の画面である。

3 プログラムコメントの生成

3.1 問題文から直接行コメントを生成する直接生成手法

ベースラインとして、問題文から直接行コメントを生成する手法について述べる。プログラムの問題文と行コメントを対訳データとして、Seq2Seqを学習させる。問題文から直接行コメントを生成する手法の概要を図3に示す。

問題文という短い文章からプログラムの各行のコメントである長い文章を生成する手法であり、英語と日本語の翻訳等とは異なり、抽象的な情報をより具体的な情報にする必要がある。そのため、生成する際に必要な情報が不足し、正しい行コメントの生成は難しいのではないかと考えられる。

3.2 段階的に行コメントを生成する2段階生成手法

問題文から直接行コメントを生成する手法の改善を目的に、ブロックコメントを間に挟むことで、行コメントを直接生成するのではなく、問題文から処理の単位ごとのコメントであるブロックコメントを生成し、ブロックコメントから更に具体的な行コメントを生成する2段階生成手法を提案する。

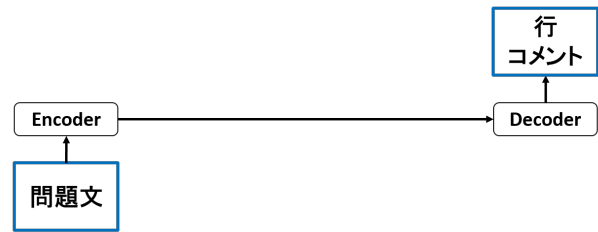


図3 問題文から直接行コメントを生成する手法の概要

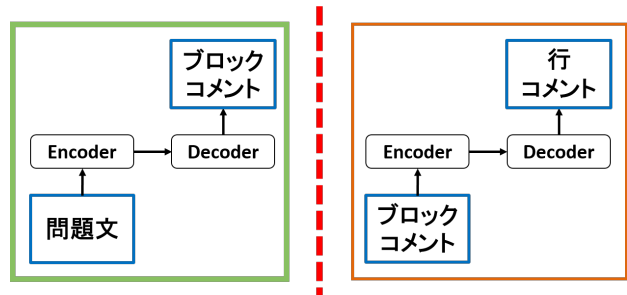


図4 2段階生成モデルの学習の概要

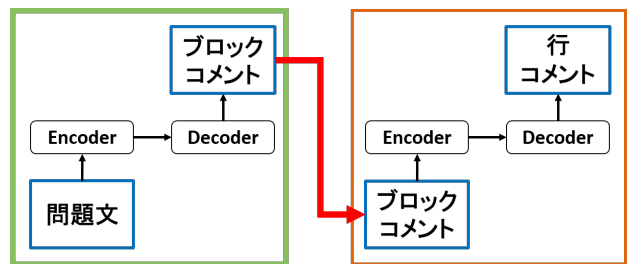


図5 2段階生成モデルの生成の概要

Seq2Seqの組み合わせ方について以下に示す。

- (1) 2段階で行コメントを生成する手法の学習の概要を図4に示す。2段階生成モデルは2つのSeq2Seqで構成するが、各Seq2Seqは独立して学習を行う。第1段階のSeq2Seqは問題文とブロックコメントを対訳データとして学習し、第2段階のSeq2Seqはブロックコメントと行コメントを対訳データとして学習する。
- (2) 2段階で行コメントを生成する手法の生成の概要を図5に示す。生成においては第1段階の学習済みSeq2Seqで問題文からブロックコメントを生成し、生成されたブロックコメントを第2段階の学習済みSeq2Seqの入力として用い、行コメントを生成する。
- (3) 2つのSeq2Seqを組み合わせ、段階的に行コメントを生成する2段階生成モデルの構造を図6に示す。第1段階では問題文の単語 x_i を順にEncoderへ入力し、変換された問題文の特徴を捉えたベクトルを用いて、Decoderはブロックコメントの単語 y_{i+1} を生成する。第2段階では

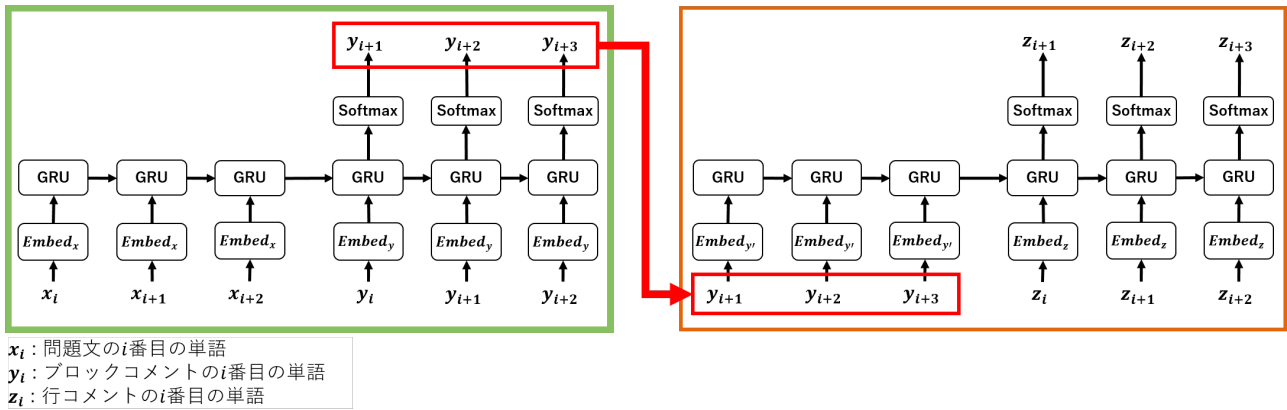


図6 段階的に行コメントを生成する2段階生成モデルの構造

第1段階で生成されたブロックコメントの単語 y_{i+1} を順に Encoder へ入力し、変換されたブロックコメントの特徴を捉えたベクトルを用いて、Decoder は行コメントの単語 z_{i+1} を生成する。

2段階生成手法では、ブロックコメントを追加しているため、モデルが利用可能な情報は増加しており、問題文から直接行コメントを生成する手法に比べ、生成される行コメントの精度が向上するのではないかと考えている。また、生成の段階を分けることで、第1段階の生成で副産物としてブロックコメントが生成される利点もある。

4 生成コメントの評価

4.1 実験環境

評価するモデルは、ベースラインとして問題文から直接行コメントを生成する直接生成モデル、問題文からブロックコメントを生成し、そのブロックコメントから行コメントを生成する2段階生成モデルの2種類である。対訳データとして、大学の情報科学科の講義で使用されたC言語のプログラム26個の問題文、ブロックコメント、行コメントを用いた。生成されたコメントは、26個のプログラムを学習と評価に9対1で分割し、オープンテストにより、評価している。評価方式については、生成されたコメントを対象に自動機械翻訳評価指標であるBLEUを用いた。

4.2 評価指標 BLEU を用いた評価

生成されたコメントのBLEUによる全体評価を表1に示す。評価に用いた3個のプログラムの問題文

表1 生成されたコメントのBLEUによる全体評価

モデル	行コメント	ブロックコメント
直接生成	0.5612	—
2段階生成	0.5871	0.8093

表2 問題文から生成されたブロックコメントの例

PRG	問題文	生成ブロックコメント
A	x と y の2つの浮動小数点を入力し、その範囲の条件にあった色を表示するプログラムを作成しなさい。	ヘッダー処理 main関数の宣言 浮動小数点型変数の宣言 入力処理 条件判定と出力関数の終了
B	2つの行列の和を計算してその行列の要素の和と最小値を表示するプログラムを作成せよ。ただし最小値の格納されている配列の添字も出力せよ。	ヘッダー処理 main関数の宣言 変数の宣言 配列の加算 $A+B \Rightarrow C$ matrixCの出力関数の終了

から生成されたブロックコメントと行コメントをBLEUで評価している。直接生成モデルはブロックコメントを生成せずに、問題文から直接行コメントを生成するため、ブロックコメントの評価は存在しない。

行コメントの評価では、直接生成モデルと比較して、2段階生成モデルはBLEUが約0.026向上した0.5871となっており、段階的にコメントを生成する2段階生成手法が有用であることが確認できる。また、2段階生成モデルの第1段階で生成されたブロックコメントの評価は0.8093となっており、高い評価であるため、行コメントの精度向上に貢献した可能性が考えられる。2段階生成モデルにおいては、第1段階の評価と第2段階の評価が比例し、第1段階の評価が低い場合は第2段階の評価も低くなるのではないかと考えている。

表3 プログラムBの行コメントの生成例と評価

問題文	直接生成		2段階生成		
	生成行コメント	BLEU	生成ブロックコメント	生成行コメント	BLEU
2つの行列の和を計算してその行列の要素の和と最小値を表示するプログラムを作成せよ。ただし最小値の格納されている配列の添字も出力せよ。	main 関数の宣言 整数型変数の宣言 整数型変数の宣言 整数型変数の宣言	0.4592	ヘッダー処理 main 関数の宣言 変数の宣言	main 関数の宣言 整数型変数の宣言 整数型変数の配列を宣言 整数型変数の配列を宣言	0.5169
	入力を促す文章を表示 数値の入力 入力を行う の 配列内の数値の入力		配列の加算 A+B ⇒ C	行の繰り返し 行内の繰り返し 配列内で加算を行う None None 文章の表示 行の繰り返し 行内の繰り返し 配列内で加算を行う None	
	内で加算を行う None None 文章の表示 行の繰り返し 行内の繰り返し 配列内の繰り返し 行内の繰り返し 配列内の繰り返し		matrixC の出力	None None 文章の表示 行の繰り返し 行内の繰り返し 配列内で加算を行う None None 文章の表示 行の繰り返し 行内の繰り返し 配列内の数値をすべて表示 None 改行 None	
	行内の繰り返し 配列内の数値をすべて None None 改行 None		関数の終了	None 関数を終了させる None	

4.3 コメントの生成例による評価

2段階生成モデルの第1段階で問題文から生成されたブロックコメントの生成例を表2に示す。プログラムAは「浮動小数点」や「入力」、「条件」、「表示」等の問題文に対応するブロックコメントを生成できている。プログラムBを確認すると、「行列の和」や「表示」等の問題文に対応するブロックコメントは生成できているが、問題文の「行列の要素の和」や「最小値」、「配列の添え字」に関するブロックコメントを生成できていない。行列の和を計算し、表示するプログラムに強く影響されたことが原因であると考えている。

プログラムBの行コメントの生成例とその評価を表3に示す。プログラムAについては直接生成モデル及び2段階生成モデル双方とも同じ行コメントを生成しており、そのBLEUも0.7270と高い評価となっている。プログラムBは変数の宣言部分では直

接生成モデルが「整数型変数」の宣言のみの生成であるのに対して、2段階生成モデルは「整数型変数の配列」の宣言も生成できている。また、プログラムAは不必要な入力処理のコメントを生成してしまっている。どちらのプログラムにおいても複数のコメントの繰り返しが発生しているが、プログラムAが理解できないコメントの繰り返しであるのに対して、プログラムBは行列の和を求める処理を連想させる一連のコメントを繰り返している。

5 まとめと今後の課題

本研究では、手続き学習システムのためのプログラムコメントの生成を目的に、ソースコードを必要としない手法として、問題文からのプログラム手続きの生成を2種類の手法で試み、段階的にコメントを具体化する2段階生成モデルの有効性が確認できた。Seq2Seqの組み合わせ方の改善やソースコードも絡めた手法の開発が今後の課題である。

参考文献

- [1]Ministry of Education, Culture, Sports, Science and Technology. Elementary school programming education guide (2nd edition). https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1403162.htm, 2018. Accessed May. 5, 2020 (in Japanese).
- [2]Ministry of Education, Culture, Sports, Science and Technology. How to programming education at elementary school level (Summary of discussion). https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm, 2016. Accessed May. 5, 2020 (in Japanese).
- [3]Haruki Kanamori, Takahito Tomoto, and Takako Akakura. Development of a Computer Programming Learning Support System Based on Reading Computer Program. In *Human Interface and the Management of Information. Information and Interaction for Learning, Culture, Collaboration and Business*, pp. 63–69, 2013. DOI:10.1007/978-3-642-39226-9_8.
- [4]Koki Okimoto, Shimpei Matsumoto, Shuichi Yamagishi, and Tomoko Kashima. Developing a source code reading tutorial system and analyzing its learning log data with multiple classification analysis. *Artificial Life and Robotics*, Vol. 22, No. 2, pp. 227–237, 2017. DOI:10.1007/s10015-017-0357-2.
- [5]Shimpei Matsumoto, Koki Okimoto, Tomoko Kashima, and Shuichi Yamagishi. Automatic Generation of C Source Code for Novice Programming Education. In *Human-Computer Interaction. Theory, Design, Development and Practice*, pp. 65–76, 2016. DOI:10.1007/978-3-319-39510-4_7.
- [6]Shimpei Matsumoto, Yusuke Hayashi, and Tsukasa Hirashima. Development of a Card Operation-Based Programming Learning System Focusing on Thinking between the Relations of Parts. *IEEJ Transactions on Electronics, Information and Systems*, Vol. 138, pp. 999–1010, 2018. DOI:10.1541/ieejieiss.138.999.
- [7]Junko Shinkai, Yoshikazu Hayase, and Isao Miyaji. A Trial of Algorithm Education Emphasizing Manual Procedures. In *Proceedings of Society for Information Technology & Teacher Education International Conference 2016*, pp. 113–118, 2016.
- [8]Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pp. 3104–3112, 2014.
- [9]Thang Luong, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, 2015. DOI:10.18653/v1/D15-1166.
- [10]Alexander M. Rush, Sumit Chopra, and Jason Weston. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 379–389, 2015. DOI:10.18653/v1/D15-1044.
- [11]Akiyoshi Takahashi, Hiromitsu Shiina, Ryunosuke Ito, and Nobuyuki Kobayashi. Procedure Generation for Algorithm Learning System using Comment Synthesis and LSTM. *International Journal of Service and Knowledge Management(IJSKM)*, Vol. 3, No. 2, pp. 48–61, 2019.
- [12]Sakuei Onishi, Akiyoshi Takahashi, Hiromitsu Shiina, 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(IJSCAI)*. to appear.
- [13]Sakuei Onishi, Fumihito Yukimoto, and Hiromitsu Shiina. Program Comment Generation Through Distributed Representation Construction Using Parse Tree Information. In *IIAI International Conference on Advanced Applied Informatics 2020*, pp. 434–439, 2020. DOI:10.1109/IIAI-AAI50415.2020.00093. Online.
- [14]Xing Hu, Ge Li, Xin Xia, David Lo, and Zhi Jin. Deep Code Comment Generation. In *2018 IEEE/ACM 26th International Conference on Program Comprehension (ICPC)*, ICPC '18, p. 200–210, 2018. DOI:10.1145/3196321.3196334.
- [15]Xing Hu, Ge Li, Xin Xia, David Lo, and Zhi Jin. Deep code comment generation with hybrid lexical and syntactical information. *Empirical Software Engineering*, Vol. 25, No. 3, pp. 2179–2217, 2019. DOI:10.1007/s10664-019-09730-9.
- [16]Boao Li, Meng Yan, Xin Xia, Xing Hu, Ge Li, and David Lo. DeepCommenter: A Deep Code Comment Generation Tool with Hybrid Lexical and Syntactical Information. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, p. 1571–1575, 2020. DOI:10.1145/3368089.3417926.
- [17]Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014. DOI:10.3115/v1/D14-1179.
- [18]Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, 2002. DOI:10.3115/1073083.1073135.
- [19]Kohei Sakane, Nobuyuki Kobayashi, Hiromitsu Shiina, and Fumio Kitagawa. Kanji Learning and Programming Support System which conjoined with a Lecture. In *IEICE Technical Report*, Vol. 114 of *ET2014–86*, pp. 7–12, 2015.