

# チャットボットを利用した プログラミング学習者に対するコーディング支援方法の提案 模範解答プログラムを利用した質問木の生成

2018SE052 武藤健太 2018SE060 西山雄也

指導教員：蜂巢吉成

## 1 はじめに

プログラミング演習において、行き詰まっている学習者はわからない箇所を教員や大学院生などのティーチングアシスタント (TA) に質問する。行き詰まり学習者とは一人で解にたどり着けない学習者を指す。教員や TA は学生に対して個別対応できる人数には限りがある。大人数に対して教員や TA のみで個別にフィードバックを行うのは困難である。質問を待機している学生は、問題解決までに時間がかかるので学習意欲が削がれてしまう。

本研究は、チャットボットを利用し、学習者が回答しやすく効率の良い質問を行う方法を提案する。学習者が回答しやすい質問とは、質問文に処理の内容や説明がわかりやすく記述されている質問のことである。学習者が回答しやすい質問を行うことで、問題解決の糸口を掴みやすくなる。質問効率を高めるとは、闇雲に質問するのではなく、必要な質問のみを記述することである。質問効率を高めることで学習者が欲する解答にいち早くたどり着くことができる。

次の3点を本研究のアプローチとする。

- 模範解答を使用し、選択肢を用いた質問を行う
- 質問ブロックを用いた質問を行う
- コメントを用いて問題ごとに特化した質問を行う

質問ブロックとはソースコードを条件分岐や繰り返し処理などの塊で捉えたものを指す。ブロックに説明を加えることで学習者がブロックでどのような処理をしているのか理解しやすくなる。不要だと判断したブロックを質問せずに飛ばすことで、全体の質問数を減らすことができるので、質問効率を上昇させることができる。コメントを用いることで定型的な質問文にならず、問題に特化した質問を行うことができる。質問文を変更したい場合、コメントを変更するだけで容易に変更できる。

## 2 関連研究

河本 [1] は、プログラムの行き詰まり解消に必要な操作の「挿入」と「更新」についてフィードバックを行う。フィードバックはコメント用いてメッセージを出力する。

木鎌 [2] は、XML によってソースコードを意味のまとまりで分け、フィードバック箇所を特定し、学習者のソースコードの状態に合わせた順番でメッセージを出力する自動フィードバック方法を提案している。本研究ではチャットボットを利用することで学習者のソースコードの記述状況

によらず、全く記述できていない学習者であっても気軽に質問できるツールの提案を行う。

渥美 [3] らは、受講学生、TA、及びチャットボットによる三者のオープンチャットと教室を巡回する移動型ロボットを利用した演習授業支援システムの開発を行っている。質問の応答は LSTM エンコーダ・デコーダ に基づいて生成され、応答文を推論して発信する。

岩崎 [4] らはチャットボットサービスの変換とその分類について述べている。現在では目的ごとに FAQ 型、雑談型、シナリオ型に分類できると述べている。既存のチャットボットシステムでは自由記述や音声入力などが提案されているが、本研究では選択肢を提示することで、学習者の解答の誘導や質問の効率化を図るツールの提案を行う。

チャットボットとはチャットに応じて自動返答する仕組みのことで、チャットボットにはシナリオに沿った回答を行う「ルールベース型」、膨大なデータを解析し、返答を行う「機械学習型」が存在する。学習者と教員、TA のシナリオはある程度予測できるので、ルールベース型のチャットボットを用いる。本研究では IBM が開発した Watson Assistant [5] を使用する。Watson Assistant には HTML を用いた埋め込みコードの自動生成が機能として備わっているため、演習問題の Web ページに埋め込んで学習者に使用させることを想定している。

## 3 チャットボットによるコーディング支援の提案

### 3.1 チャットボットによるコーディング支援の概要

本研究では模範解答をもとにチャットボットが質問する学習支援方法を提案する。この学習支援はプログラミング教育の演習時や自習時に行う。チャットボットからの質問で学習者に問題のヒントを与えることを想定している。模範解答から質問を生成することで、問題ごとに特化した質問を行うことができる。基本的なやりとりのみ行い、それ以外の質問は教員や TA に対応を促す。

質問はチャットボット側から行い、「○○する は記述しましたか」のように質問する。○○には模範解答に記述されたコメントが入り、には模範解答から抽出した関数や制御文などが入る。質問は選択肢を用いて行い、関数や制御文に着目した質問を行う。学習者は「はい」、「いいえ」の二者択一の回答をすることを想定している。「はい」、「いいえ」を用いる理由は、学習者の回答範囲を絞ることで回答しやすくするためである。チャットボットから

の質問によって学習者が悩んでいる原因が解明された場合、悩んでいる内容に合わせたフィードバックメッセージをチャットボットが出力する。フィードバックを出力後、再びブロック分けから生成した選択肢を提示する。これにより、学習者はソースコード完成まで何度も利用することが可能である。

### 3.2 前提条件

模範解答の関数や制御文に着目した質問を行い、それに伴うコメントの記述は教員が行うものとする。コメントはその行で何をするかを記述し、関数や制御文と同行に記述する。必要であれば、変数宣言や代入文と同行にもコメントを記述する。模範解答は 1 行につき 1 文の記述とする。2 章で述べた通り、本研究では全く記述できていない学習者も対象とするので、学習者のソースコードと模範解答の比較は行わない。質問対象の関数は 1 つのみとし、main 関数を対象とする場合は main 関数のみ質問シナリオが生成される。if, for, while 等の制御文の記述を行う際、波括弧は省略せずに記述するものとする。基本的な質問のみフィードバックを行い、答えるのが難しい問題は教員や TA に対応してもらう。

### 3.3 模範解答からの質問木の生成

質問木は、質問同士のノードの繋がりを表した木構造である。チャットボットにダイアログを組み込む前段階として、質問の流れを決定する質問木の生成を行う。

質問木の生成にあたって次の課題がある。

- 学習者が回答しやすい質問を行う。
- 学習者への質問を最小限に抑えることで質問効率を高める。

前者の解決方法として、模範解答に記述したコメントを補足説明とする質問方法を提案する。質問方法として例えば「○○する if 文は記述しましたか」と質問する。○○にコメントを記述することで処理の内容を踏まえた質問を行うことができるので、どの箇所について聞かれているのかの状況把握に繋がる。

後者の解決方法として、選択肢を用いる方法を提案する。最初に選択肢をいくつか提示し、聞きたい箇所に近い選択肢を学習者に選択してもらうことで必要な箇所のみ質問を行うことができる。

模範解答から質問木の生成までのアルゴリズムについては次のようになる。

- 模範解答を構文解析し、構文木に変換する。
- 構文木について幅優先探索を行い、選択肢を生成する。
- 構文木について深さ優先探索を行い、質問木を生成する。

これらの課題解決方法やアルゴリズムをもとに質問木を生成することで、学習者にわかりやすく効率の良い支援が可能になると考える。

例として、平均の計算の模範解答をソースコード 1、生成される質問木を図 1 に示す。ノードは 4 種類、エッジは 2 種類ある。破線の楕円は質問されないノード、二重線の八角形は選択肢ノード、実線の楕円は質問文ノード、四角形はフィードバックメッセージノード、実線のエッジは学習者の回答、破線のエッジはシステム上の遷移をそれぞれ表す。表示の都合上、質問の開始と終了のノードは同じノードだが 2 つに分類している。

ソースコード 1 平均の計算 模範解答

```

1 double average ( int arr [ ] , int size ){
2     // +@ 変数宣言
3     int i ; // 繰り返し用の変数を宣言
4     double sum ; // 合計の変数を宣言
5     double avg ; // 平均の変数を宣言
6     // -@
7     // +@ 計算方法
8     sum =0; // 合計を求め変数を初期化 +& 初期値 0で初期化
9     for ( i =0; i < size ; i ++){ // データの数の分繰り返し +& 配列の先頭から最後まで値を1ずつ増やしてデータの数の分繰り返し
10        sum = sum + arr[i]; // 合計の計算 +& 合計の変数に配列を累積加算
11    }
12    avg = sum / size ; // 平均の計算 +& 合計とデータ数を用いて平均を計算
13    // -@
14    return avg ; // 平均の値
15 }
```

### 3.4 質問内容の記述

教員は学習者に質問したい制御文や関数、変数宣言と同じ行に質問内容を記述する必要がある。実際にソースコード 1 を用いて説明する。ソースコード 1 の 3 行目のように同じ行内にコメントの記述が 1 つの場合、質問文とほぼ同じ文章をフィードバックすることになる。しかし、コメントを 2 つ記述することでフィードバックメッセージの内容を設定することができる。8 行目では「合計を求め変数を初期化」と「初期値 0 で初期化」の 2 つのコメントが記述してある。このように 2 つのコメントを模範解答に記述する場合、「+&」をコメント間に記述する。1 つ目のコメントに質問文を記述することで学習者の悩んでいる箇所を特定し、2 つ目のコメントを説明文として記述することで、行き詰まりを解消するメッセージを表示する。コメントを全く記述していない場合は質問を行わない。

### 3.5 質問ブロックの概要と分類

図 2 はソースコード 1 のブロック化のイメージ図である。例えば、学習者が for 文周辺で悩んでいる場合、下のようにブロック A、ブロック B、ブロック C と処理を塊で分けることで学習者が聞きたい for 文周辺の箇所のみピンポイントで質問することができる。質問ブロックを用いる理由としてソースコードを複数の処理の塊で捉えた方が学習に繋がり、必要な質問ブロックのみ聞くことで質問効率

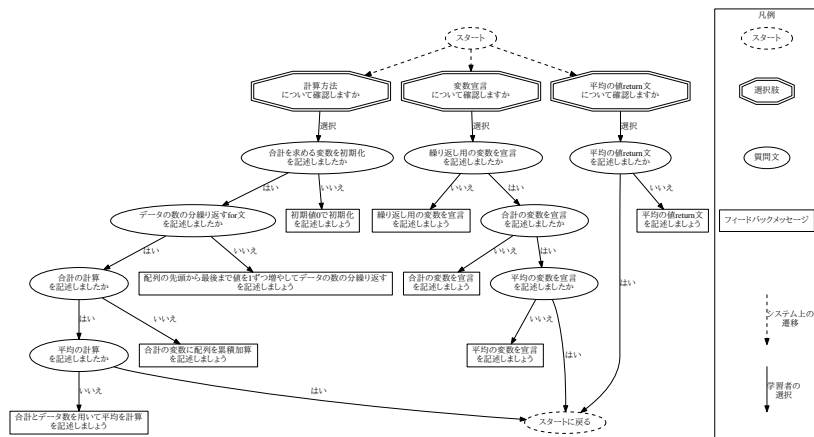


図1 ソースコード1の質問木

が上がると思ったためである。

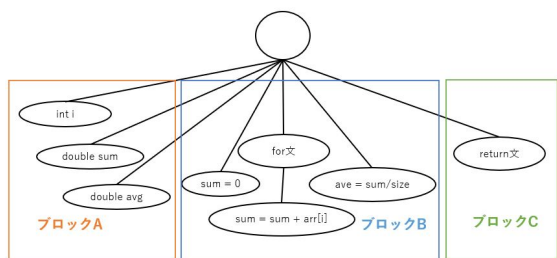


図2 ブロック化のイメージ図

質問ブロックの分け方として手動、自動、手動と自動の両方で分類する方法がある。基本的には自動で分類を行い、質問ブロックが生成される。しかし、教員が意図した箇所に質問ブロックを生成できるように手動での分類について考えた。自動での質問ブロックの分類は制御文をもとに分類し、制御文の始まりから次の制御文の始まりまでが1つの質問ブロックとなる。手動と自動での質問ブロックの分類は手動で分類した箇所以外を自動分類する。手動のみで分類したい場合、記号を用いて分類する。質問ブロックを分類することで選択肢が生成される。その基準は以下ようになる。

選択肢になる基準

- 最初のコメント
  - 「+@」を含むコメント
  - 「-@」の次のコメント
  - 1番上位の制御文
- 選択肢に戻る基準
- 最後のコメント
  - 「-@」を含むコメント
  - 次のコメントが選択肢

手動での質問ブロックの分類はソースコードに「@」を

コメントで記述する。「+@」をコメントに記述した場合、その行が質問ブロックの開始位置となる。「+@」を記述した行がコメントのみの場合、「+@」以外のコメントが質問ブロックの説明文となる。ソースコード1の2行目と7行目にはコメントのみ記述されているので「変数宣言」と「計算方法」がそれぞれの質問ブロックの説明文となり、Watson Assistantで選択肢としてのみ表示され、質問はされない。「-@」をコメントに記述した場合、その行が質問ブロックの終了位置となる。質問ブロックとなる行が1行のみの場合は、「+@」と「-@」の両方をコメントとしてその行に記述する。「+@」と「-@」で囲われていない箇所は自動でブロック分けされる。したがって、「計算方法」「変数宣言」「平均の値 return 文」が最初の選択肢になる。

### 3.6 設計と実現

本研究では質問同士のノードのつながりを表すグラフの作成や Watson Assistant に必要なダイアログの作成を自動化するシステムを開発する。ダイアログの作成を自動化するまでに3つの段階が存在する。

- 1 段階 模範解答を構文木へ変換する。
- 2 段階 構文木から抽出したデータから質問木を生成する。
- 3 段階 質問木からグラフの生成や Watson Assistant に必要なダイアログを生成する。

模範解答から Watson Assistant に必要なダイアログの作成までを自動化するツールを開発する。3.3 で述べたように模範解答から質問木を生成し、その質問木からダイアログが生成する。学習者とチャットボットのやり取りは図3のように行う。提示された選択肢を学習者がクリックすることで次のノードに移る。「はい」をクリックした場合、次の質問へ移動し、「いいえ」をクリックした場合、その質問に対するフィードバックメッセージが表示される。

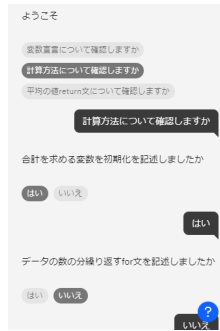


図3 学生とチャットボットの会話の例

## 4 評価・考察

表1は模範解答の上から質問を行った場合と選択肢を用いて質問を行った場合の質問数を比較した結果である。前者の平均質問数4.5, 最大質問数8, 後者の平均質問数3.1(小数点第二位四捨五入), 最大質問数5となる。選択肢を用いた方が平均質問数, 最大質問数が削減されているので, 質問の効率化を図れている。

表1 質問数の比較

ノード名	上から質問した場合の質問数	選択肢を利用した場合の質問数	結果
繰り返し用の変数を宣言	1	2	多い(選択肢)
合計の変数を宣言	2	3	多い(選択肢)
平均の変数を宣言	3	4	多い(選択肢)
合計求める変数を初期化	4	2	少ない(選択肢)
データの数の分繰り返し for 文	5	3	少ない(選択肢)
合計の計算	6	4	少ない(選択肢)
平均の計算	7	5	少ない(選択肢)
平均の値 return 文	8	2	少ない(選択肢)

しかし, 質問の効率化が図れない場合がある。ソースコード2は行列の積の計算を行う模範解答である。先ほどと同様に質問数を比較すると, 上から順番に質問を行った場合の平均質問数3.5, 最大質問数6, 選択肢を用いた質問を行った場合の平均質問数3.7, 最大質問数6である。最大質問数は変わらないが, 平均質問数は前者の方が少ないため, 選択肢を用いても効率化できていない。このような結果になる理由は, 選択肢の生成位置が原因である。最初の2つの選択肢が上から1つ目と2つ目のコメントから生成されている。したがって, 選択肢を用いてもスキップできる質問数が少ない。このような問題を解決する方法として, 選択肢の生成数を増加するか, 生成位置を分散する方法がある。

### ソースコード2 行列の積の計算 模範解答

```

1 void MatMult(int a[2][2],int b[2][2],int c
  [2][2]){
2   int i; // 変数宣言 +& 必要な添字の変数宣言
3   int j;
4   int k;
5   // +@ 行列の積の計算
6   for(i=0;i<2;i=i+1){ // 行を数える +& 行数0
   から2未満まで繰り返す
7     for(j=0;j<2;j=j+1){ // 列を数える +& 列
   数0から2未満まで繰り返す

```

```

8     c[i][j] = 0; // 要素の初期化 +& 要素
   を0で初期化
9     for(k=0;k<2;k++){ // 一方の行ともう一
   方の列を数える +& 行数と列数を同時
   に0から2未満まで繰り返す
10    c[i][j] = c[i][j] + a[i][k] * b
   [k][j]; // 計算式 +& 対象の
   要素に行列の積を加算
11    }
12  }
13 }
14 // -@
15 }

```

## 5 おわりに

本研究では, 多数の行き詰まりプログラミング学習者への対応や教員の負担を減らすことを目的に, チャットボットを用いて効率よくわかりやすい質問を行う教育支援ツールについて提案した。グラフの作成にはオープンソースの Graphviz を利用し, 質問木を可視化する。様々なパターンの質問木を比較した結果, 一部の例外はあるものの, 質問木の2つ目以降の質問については, ほとんどの場合で質問の効率化を図ることができていることを確認した。今後の課題として, 一部効率化を行っていない質問木の対応や記述ミスを正解だと判断するなどのトラブルへの対応, 実際の学習に適用した評価が挙げられる。

## 参考文献

- [1] 河本菜々: “プログラミング学習におけるソースコード文に対するフィードバックメッセージ生成方法の提案”, 南山大学理工学部 2020 年度卒業論文 (2021).
- [2] 木鎌汐里: “学習者の編集途中のソースコードに対する自動フィードバック方法の提案—処理単位に基づいたフィードバックブロックの定義—”, 南山大学理工学部 2020 年度卒業論文 (2021).
- [3] 渥美雅保, 村田祐樹, 安川葵: “SOTARO:オープンチャットとロボットの連携による演習授業支援システム—Human-in-the-loop 型機械学習によるアプローチ—”, 情報処理学会第 79 回全国大会, 2017.
- [4] 岩崎信也, 津村直也: “チャットボットサービスの変遷とそれを支える構成技術—シナリオ型チャットボットサービスの発展—”, 情報処理学会論文誌, Vol.62 No.10 2021.
- [5] IBM Watson Assistant-概要-日本, <https://www.ibm.com/jp-ja/cloud/watson-assistant>, (参照 2021-07-18).
- [6] 吉田敦, 蜂巣吉成, 沢田篤史, 張漢明, 野呂昌満: 『属性付き字句系列に基づくソースコード書き換え支援環境』, 情報処理学会論文誌, Vol.53, No.7, pp.1832-1849, 2012.
- [7] Graphviz, <https://graphviz.org>, (参照 2021-09-25).