

チャットでの質問履歴から推定する習熟度を考慮した プログラミングヒント提示システムの開発

平野 翔也, 鷹野 孝典
神奈川工科大学 情報学部 情報工学科

Development of a Programming Hint Presentation System according to Learner's Proficiency Estimated from Question History by Chat

Shoya Hirano, Kosuke Takano
Department of Information and Computer Sciences,
Faculty of Information Technology, Kanagawa Institute of Technology

This study presents a hint information presentation system for supporting the programming learning according to the learner's proficiency. The chat function is incorporated into our system, and a learner can get the knowledge of programming through communications with a teacher in addition to the hint information of programming that the proposed system provides. The feature of our system is to estimate the proficiency level of programming for each learner from the history of questions by chatting with the teacher, and provide the hint information according to the learner's understanding level. In the experiment of classifying a set of questions into some groups according to the understanding level, we confirm the feasibility of the proposed system.

キーワード: プログラミング言語教育, プログラミング学習, 習熟度推定, 学習チャット

1. はじめに

マルチメディアや Web 技術の進展により, オンラインで海外や遠方の教育機関の講義を受講できる MOOC (Massive Open Online Course) ⁽⁸⁾⁽⁹⁾ が普及し, 多くの受講者を獲得している. また, スマートフォンやタブレットで学習できる音や視覚効果に富んだ学習教材も数多く存在し, 個人学習や授業の補助教材として活用されている. このように, 教育形態も多種多様に変化してきおり, 学習者のレベルに応じた学習方法や指導方法の需要が高まっている.

例えば, MOOC では, ビデオ講義の中で問題や疑問に対するヒントを提示し, 問題を解くための過程を考える機会を作ることで, 更なる学習効果の向上が見込められると思われる. しかし, MOOC では受講生数が大規

模であるため, 個々の学生のレベルに応じた回答やヒントを与えることは困難である. また, 学習者が個人学習をする場合において, 適切なレベルの解説やヒントが提供されないと学習が滞ってしまう可能性がある. 一方, プログラミング教育では, 例題となるサンプルプログラムや解説をヒントとしながら, 学生に演習させる機会が多い. この際, プログラミング学習者の習熟度が推定できれば, その人の学力レベルに合ったヒントの提示を行えると考えられる. 既存の手法では, 学習履歴や成績から学習者の習熟度を推定することが多い. しかし学習履歴や成績だけでは, 例えば, 新しく別のプログラミング言語を学び始めた学習者の, 対象となるプログラミング言語の習熟度を推定することは困難な場合がある.

本研究では, プログラミング学習において, 教師と

のチャットを通じた質疑応答が可能な、プログラミング学習支援システムを提案する。システムの特徴は、チャットでの質問履歴から学習者のプログラミング習熟度を推定し、学習者のレベルに応じたプログラミングに関するヒント情報を提示する点にある。実験では質問文の習熟度ごとの分類精度を検証し、提案システムの実現可能性を検証する。

2. 関連研究

チャットでの自動応答やチャットを利用したプログラミング学習に関して多くの研究がなされている(1)(2)(3)。文献(1)では、ペアプログラミングにおいてテキスト対話を用いた教育手法を提案しており、対話（チャット）の頻度はコーディングやエラーの修正時間に影響すると報告している。安川らは、チャットでの学生の発言について人が対応すべきか、ボットでの対応で十分かを機械学習により判定する手法の考案と性能評価を行った(2)。また、平井らは、プログラミング学習におけるペアプログラミングの成功事例と失敗事例にみられる会話の違いを比較・分析している(3)。

一方、個々の学習の理解度等に応じた学習環境を実現するために、ユーザ情報抽出や習熟度推定に関する手法も数多く提案されている(4)(5)(6)(7)。文献(4)では、パーソナライズ可能な対話システムのために、ユーザ発言からユーザ自身に関する情報を構造化された形で抽出する手法を提案している。文献(5)では、誰がどのアイテムを読んだかという関係からなる2部ネットワークの情報のみを用いてアイテム難易度とユーザ習熟度を推定するアルゴリズムを示した。また、小田切らは、採点結果などをもとに利用者の苦手度を算出し、問題の出題傾向や指摘内容に反映させる敬語学習システムを通してより実践的な敬語学習を試みた(6)。文献(7)において、三好らは、ユーザの習熟度と学習コンテンツの難易度のマッチングを考慮に入れたコンテンツ推薦手法の開発を目的として、習熟度と難易度の推定手法について示した。

本研究では、プログラミング学習者の習熟度に応じた質問に関するヒント情報を学習者、教師の両者に提示することで、学習者には問題の解法や疑問解消への手助け、教師には解答や教え方に工夫を加えることが

できる学習チャットシステムの実現を目指している(図1)。



図1 学習チャット機能の画面設計

3. 提案システム

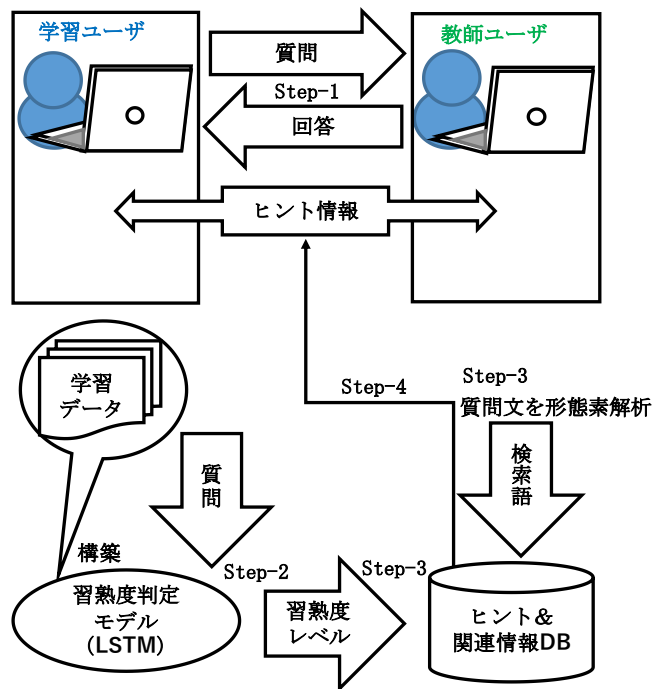


図2：提案方式の概要図

3.1 提案システムの概要

提案システムの概要図を図2に示す。提案システムは、プログラミング学習において、チャットを通じた学習者の質問履歴から学習者のプログラミング習熟度を推定し、習熟度に応じたプログラミング学習に関するヒントを学習者・教師の両方に提示する。提案システムにおける学習者の習熟度判定およびヒント提示は、下記の手順で実施される。

Step1: 学習ユーザは教師ユーザに質問を行い、教師ユーザは回答を送信する。

Step-2 : 質問文を習熟度判定モデルに入力し、習熟度レベル (N 段階) を得る。質問文の履歴から、各質問文に対応する習熟度レベルの平均値を習熟度 P として算出する。

Step-3 : 質問文を形態素解析で分割し、名詞を抽出する。Step-2 で算出した習熟度 P と抽出した名詞単語を入力として、ヒント情報データベースへ問い合わせる。

Step-4: Step-3 で問い合わせ結果として得られたヒント情報を学習ユーザ、教師ユーザの両方の画面に表示する。

3.2 習熟度判定モデルの構築

質問文から習熟度を推定するために、質問に含まれる単語、質問の言い回し、質問文の長さといった特徴が利用できる。本研究では、これらの特徴を表現学習するために深層学習による LSTM (Long short-term memory) を適用した深層学習を行い、習熟度判定モデルを構築する。習熟度判定モデルを構築するために、各質問文に対して習熟度に関する正解ラベル (習熟度ラベル) を付与する必要がある。例えば、プログラミング学習時の質問文に含まれる専門用語を学習項目ごとに分類し、さらに初級者が着目しやすい学習項目、上級者が着目しやすい項目などに対応付けて分類することで、各質問文に対する習熟度ラベルを付与することができる (表 1)

表 1 学習項目に応じた習熟度設定の例

学習項目	習熟度ラベル / 質問文の例
<ul style="list-style-type: none"> 変数 分岐, 繰り返し 関数 	LEVEL-1 (初心者) / 「〇〇するプログラムを教えてください」
<ul style="list-style-type: none"> クラス, 構造体 ファイル操作 GUI 画像処理 	LEVEL-2 (中級者) / 「クラス継承の仕方がわかりません」
<ul style="list-style-type: none"> データ分析 API フレームワーク 人工知能 Raspberry pi 機械学習 	LEVEL-3 (上級者) / 「python の Pandas データフレームから, 所定の辞書型に変換する方法を教えてください」

3.3 習熟度の算出

3.1 節の Step-2 における習熟度 P の算出手法について述べる。チャット履歴において、 N 個の質問文が含まれるとする。この場合、学習者の習熟度 P は、各質問文から判定される習熟度レベル (初級者レベル=1, 中級者レベル=2, 上級者レベル=3, ...) を重みスコアとした総和平均として、下記式により算出する。

$$\text{習熟度}P = \frac{L1 \times S1 + L2 \times S2 + \dots + Ln \times Sn}{\text{質問総数}N}$$

ここで、 Ln は LEVEL- n の質問数、 Sn は LEVEL- n の重みスコアとする。例えば、15 個の質問文のうち、初級者、中級者、上級者と判定された質問文の数がそれぞれ、5 個、4 個、6 個とすると、習熟度 P は下記のように計算される。

$$\begin{aligned} \text{習熟度}P &= \frac{5 \times 1 + 4 \times 2 + 6 \times 3}{15} \\ &= 2.06666 \dots \approx 2 \text{ (LEVEL2)} \end{aligned}$$

3.4 質問文に対するヒント情報の提示

提案システムでは、学習者の習熟度に応じて、質問文に対するヒント情報を、チャットをしている学習者と教師の両方に提示する。学習者に対しては、質問から習熟度を考慮することで、その人の理解に沿ったヒント情報を提示し、解答を導く手助けをすることができる。また、教師としても質問する回答の補助資料として Web 上でチュートリアル記事などを学習者に提示する機会も増えており、そのような補助資料を調べる手間を減らすとともに、習熟度の把握によって指導内容や教え方を工夫することができる。

ここで、習熟度が 1~3 の場合に対応する質問及びヒント情報の例を表 2 に示す。

表 2 習熟度に応じたヒント情報の例

習熟度	質問	ヒント情報		
		左側に配列名を書き、右側で初期化します。	配列への追加は append(値)です。	要素の挿入は insert(位置, 値)です。
1	python の配列はどう書けばいいですか？			
2	python の GUI ツールである Tkinter でのボタンの作り方がわかりません	Tkinter でのボタンは .Button と書く必要があります。	ボタンの名前には text='名前' で変更できます。	チェックボックスやラベルもボタンと同じ書き方で作ります。
3	python の機械学習で分類が複数のとき、model.compile は、どの損失関数を書けばいいですか	複数の場合は categorical が必要です。	optimize r は adam がよく利用されているようです。	モデル構築の最後の活性化関数は softmax がいいでしょう。

4. 実験

プログラミング学習に関する質問文を対象として習熟度に応じた分類精度を評価し、提案システムの実現可能性を確認する。

4.1 実験環境

プログラミングに関する質問文を yahoo 知恵袋⁽¹⁰⁾, Teratail⁽¹¹⁾, stackoverflow⁽¹²⁾の3つの Q&A サイトから収集した。学習データ、テストデータとして、それぞれ 335 件、135 件用意した。また、質問文の内容から表 1 に基づいて分類し、習熟度ラベルを設定した。分類数の内訳を表 3 に示す。

表 3 質問文の件数

	初心者	中級者	上級者
学習用 : 335 件	110 件	116 件	109 件
テスト用 : 135 件	45 件	45 件	45 件

4.2 実験方法

表 3 に示す 335 件の質問文・学習データを LSTM を用いて学習し、習熟度判定モデルを構築する。135 件の質問文・テストデータを習熟度判定モデルに入力し、習熟度に関する分類精度を算出する。ここで構築した習熟度判定モデルの学習精度は約 38%であった。

4.3 実験結果

分類結果として 40%の分類精度が得られた。分類の成功例と失敗例の結果を表 4 に示す。

正解ラベルは表 1 に示したように学習項目に応じて付与した。例えば、「機械学習」に関する質問は正解ラベルを習熟度 3(LEVEL3)とし、「繰り返し」に関する質問は習熟度 1(LEVEL1)とした。表 4 の結果において、習熟度判定が成功した質問文は該当する習熟度に沿った項目に関する内容であるため、正解したと考えられる。一方、不正解となった質問文については、構築した習熟度判定モデルでは、「numpy」や「opencv」で使用する関数名に関する単語が学習しきれておらず、それらが影響したことも考えられる。また、習熟度が高いほど質問文が長めとなり、内容が詳しい、かつ質問文中の言い回しが丁寧である傾向が見受けられた。このような特徴も考慮して、習熟度判定モデルを構築することで分類精度の向上が期待できると考えている。

表 4 習熟度判定の成功例と失敗例

習熟度		質問	P
1	成功例	「python 繰り返し処理で出した結果を連続して結合させる方法を教えてください。」	1
	失敗例	「Numpy を用いれば行列の計算が出来ることがしていますが使わなければならないのでしょうか？用いずに計算する方法があれば知りたいです。」	2
2	成功例	「Python 初心者です。標準化のプログラムを作成しています。1. ファイル名を標準入力で受け取り 2. ファイルの各列を標準化 3. 結果を新しい csv ファイルに出力といった流れを考えています。エラーの意味が分からず、解決策をご提示いただけないでしょうか。何卒、よろしく願いいたします。」	2
	失敗例	「python の opencv 円検出の精度を上げる方法をおしえてください。」	3
3	成功例	「当方 python 初心者です。python を用いて Twitter のプロフィール文から 2 クラス分類をする識別機を作成したいと考えています。現在教師データとラベルをライブラリに引き渡して機械学習をさせようとしている段階なのですが、エラーが発生しており困っています。詳しい方どうかお力添えお願い致します。」	3
	失敗例	「keras でモデルを構築し、学習させています。カスタム損失関数を作るため、何個目のデータを読み込んでいるかを知りたいです。つまり、欲しい値は、1/90000 の 1 や 29/90000 の 29 の値です。この値はどのように取得すれば良いでしょうか。」	1

5. おわりに.

本研究では、チャット上での質問からその人の習熟度を推定する手法の検討及び習熟度を考慮したヒントを提示する学習チャットシステムを提案した。実験において、質問文に対して習熟度判定を行ったところ、分類精度は 40%となり、習熟度判定モデルの分類性能に改善の余地があることがわかった。

今後の予定として、習熟度判定手法の精度向上のため、学習データを更に増やして評価実験を実施する予定である。また、習熟度に基づいたヒント提示機能の実装とともに、提案システムのユーザビリティや有用性の評価も行っていく予定である。

参 考 文 献

- (1) 佐々木孝輔, 林勇吾, 井上智雄: “テキストチャットによるペアプログラミング学習システム: 学習者同士のインタラクションに着目した検討”, 情報処理学会研究報告ヒューマンコンピュータインタラクション, 2014-HCI-157, 53号, pp.1-8, (2014)
- (2) 安川葵, 渥美雅保「ソフトウェア演習支援のためのチャットシステムへの機械学習の組み込みと授業での運用評価」, 情報処理学会第79回全国大会講演論文集, 2017巻, 1号, pp677-678, (2017)
- (3) 平井佑樹, 井上智雄: “ペアプログラミング学習における状態の推定——つまずきの解決の成功と失敗に見られる会話の違い”, 情報処理学会論文誌, 53巻, 1号, pp.72-80, (2012)
- (4) 平野徹, 小林のぞみ, 東中竜一郎, 牧野俊朗, 松尾義博: “パーソナライズ可能な対話システムのためのユーザ情報抽出”, 人工知能学会論文誌, 31巻, 1号, pp.1-10, (2016)
- (5) 濱田一伸, 三好康夫, 鈴木一弘, 塩田健一: “2部ネットワーク分析によるユーザ習熟度とアイテム難易度の推定アルゴリズムの提案”, 教育システム情報学会第36回全国大会講演論文集, pp.384-385, (2012)
- (6) 小田切直, 樽松理樹, 藤田ハミド: “習熟度を考慮した敬語学習支援システムの開発”, 情報処理学会第68回全国大会講演論文集, 68巻, 4号, pp.637-638, (2006)
- (7) 三好康夫, 濱田一伸, 鈴木一弘, 塩田研一, 岡本竜, 金西計英: “学習コンテンツ推薦に向けた2部ネットワーク分析に基づく習熟度と難易度の推定手法の提案”, ARG Web インテリジェンスとインタラクション研究会, 1号, (2012)
- (8) Coursera,
<https://ja.coursera.org/> (2019年12月11日確認)
- (9) edX,
<https://www.edx.org/> (2019年12月11日確認)
- (10) yahoo 知恵袋,
<https://chiebukuro.yahoo.co.jp/> (2019年12月11日確認)
- (11) Teratail,
<https://teratail.com/> (2019年12月11日確認)
- (12) stackoverflow,
<https://ja.stackoverflow.com/> (2019年12月11日確認)