

PPL システムにおける偽答生成機能の開発

上倉諒佑, 小久保証宏, 布広永示

東京情報大学

Development of incorrect answer generation function in PPL system

Ryosuke KAMIKURA, Masahiro KOKUBO, Eiji NUNOHIRO*

Tokyo University of Information Sciences*

プログラミング教育の目的は、プログラム言語の文法や構文要素などの知識の習得だけではなく、プログラミングを通して問題を詳細化し、解決策を提案する問題解決能力を向上させることであると考えられる。本研究では、学習者に対し「システム開発」に関わる要素を含んだ問題を与え、段階的詳細化を図ることによってプログラミング教育の到達目標として必要な問題解決能力を向上させる学習を支援するプログラミング学習支援システム「PPL システム (Problem based Programming Learning system using stepwise refinement)」を開発している。PPL システムでは、「システム開発」を段階的に詳細化していく5つの設問を解答する。本発表では、PPL システムの概要について解説し、Word2vec を用いて正答のキーワードから偽答を自動生成する偽答生成機能について述べる。

キーワード: 学習支援システム, プログラミング, 段階的詳細化, Word2Vec, 自動生成

1. はじめに

情報化社会の拡大により、小学校教育課程でプログラミング教育が必修化される方針が採られるといった情報処理教育に対する取り組みが活発に行われるようになってきている[1]。プログラミング学習に対してのアプローチは様々な形態があり、言語に対する知識を要さずに感覚的にプログラミングの構造を理解することの出来るビジュアルプログラミングなどが中心となってプログラミングの初学者に対して用いられてきている。しかし、これらの取り組みはコーディングという工程の流れを簡略化して学習させているだけに過ぎない。

本研究では、ウォーターフォールモデルを適用したシステム開発における問題の段階的詳細化をプログラミング教育に取り入れた。そして、プログラミングを行う必要のある機能部分や処理の流れを理解することを重視し、プログラミングの工程の前段階として問題発見・解決の手順を理解するための学習を支援するプログラミング学習支援システムを開発した。本発表では、問題中にある択一問題作成において、正答のキー

ワードを元にして偽答を自動生成することにより問題作成する管理者側に必要な問題入力工程を効率化する機能を提案する。

2. PPL システム概要

2.1 設計方針

本システムは、(1) 初学者に対して言語知識やプログラミングの構造知識を要さない日本語ベースの問題を与える、(2) マウス操作に特化する、(3) 段階的詳細化により問題をそれぞれのプロセスに分割して捉えさせる、(4) 出題者の問題追加の負荷軽減を重視するという4つの設計方針で開発している。

(1) の目的は、プログラミングの前工程である問題の理解と読み取りに重きをおいた内容となっておりプログラミング言語の構文要素などを出さないことで学習者の知識レベルを問わないことである。

(2) の目的は、キーボード入力を行わずマウス操作で解答することで、様々な環境で学習を可能とし、更にタブレットなどの場合キーボード入力の大半がタッチスクリーンになってしまい解答のレスポンスが悪

くなる問題を解決することである。

(3)の目的は、PPLの問題の各設問に対して以下のような段階のプロセスを与え、設問の役割を明確にすることである。

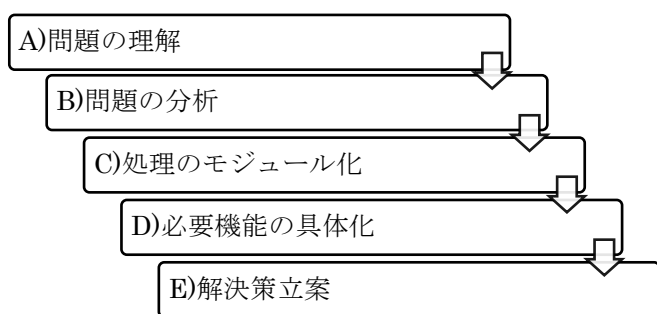


図 1 PPL システムの解答プロセス

A) 問題の理解

問題文から、利用者の操作手順や処理に必要なデータについて考える。設計工程における外部設計である。

B) 問題の分析

利用者の操作手順と操作に対応したシステムの機能について考える。設計工程における内部設計である。

C) 処理のモジュール化

機能毎に、処理能力と入出力情報を考える。設計工程における内部設計である。

D) 必要機能の具体化

モジュール毎に処理内容、入力として必要な情報、出力されるべき情報を考える。設計工程における内部設計である。

E) 解決策の立案

提案した機能を用いてシステムの側面から処理の流れをチャートで表現する。設計工程におけるプログラム設計である。

A)～E)を段階的に解答していこことで前工程の重要性を理解する。

(4)の目的は、問題の出題者が新規の問題を追加する際に必要最低限の文章、項目を入力することで問題生成時の出題者の負担を軽減することである。

2.2 システムの構成

本システムは、学習者操作系、自動処理系、出題者操作系の3系統により構成されている。本システムの構成を図2に示す。また、図3に本システムの利用の流れを示す。

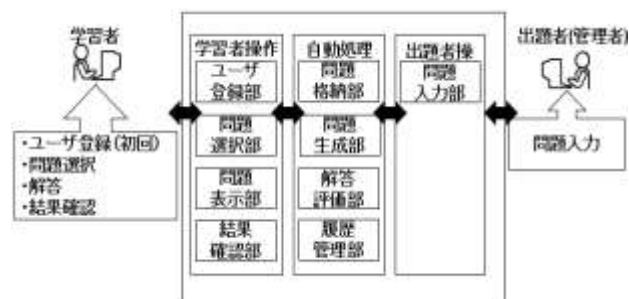


図 2 システムの構成

2.3 利用者のアクセスから終了までの動作概要

学習者と本システムの利用の流れを次に記述する。

- ① 学習者はログイン画面にアクセスする。
- ② 初回利用時はログイン画面からユーザ登録画面に移行してIDとパスワードを登録する。
- ③ ログイン後に問題選択画面にある問題群から解答する問題を選択する。
システムは、学習者に選択した問題の表示に必要な情報群をデータベースから取得する。
- ④ 学習者は、設問を1問ごとに正解するまで次の設問には進まずに解答を行う。設問内の問題をすべて正答した時、次の設問に進む。

各設問の内容は以下の通り。

設問1：択一形式問題

設問2：文字列ピースの並び替え

設問3：機能の概要に対するピース並び替え

設問4：機能の入出力に対するピース並び替え

設問5：チャートの並び替え

- ⑤ 設問5まで正答したら、解答にかかった時間・不正解の数をもとに成績のランクを解答結果画面にて表示する。
- ⑥ 各画面から前回の成績データを確認出来る。
- ⑦ 問題への挑戦をやめる場合などは各画面内に表示されているログアウトボタンを選択することで利用を終了する。

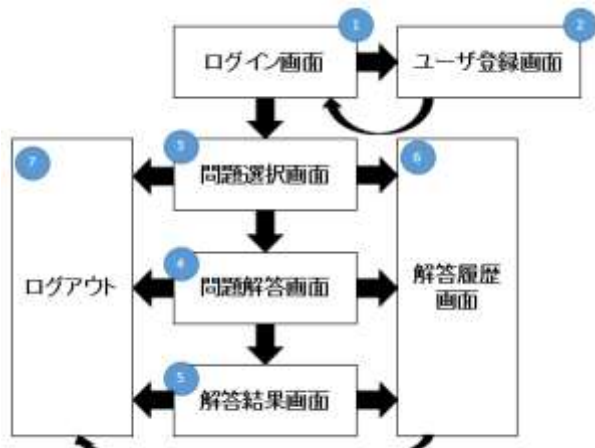


図 3 システム利用の流れ



図 4 解答画面設問 1 の例

3. 問題の生成

3.1 問題の入力手順

問題入力画面の例と追加手順を次に示す。

問題番号:

タイトル入力:

(a) 問題文入力:

【設問1で使われる部分です】
 (b) 操作部分:
 3

【設問2で使われる部分です】
 (c) 主制御部分(正解の並びで入力):

【設問3で使われる部分です】
 (e-1) 機能部分(ピースは正解の並びで入力してください):

【設問4で使われる部分です】
 (e-2) 機能部分:

【設問5で使われる部分です】
 (f-1) 全体チャート:
 パーツ追加:

図 5 問題入力画面

入力画面では各設問の構成に必要な入力フォームとボタンが表示されている。

- ①問題番号を入力する
- ②問題タイトルを入力する
- ③問題文を入力する
- ④設問 1 の内容を入力する。

設問 1 は操作関連の問題で文章と択一問題で構成される。文章入力フォームと選択問題の「正答」を入力する。設問 1 の出題では、偽答を Word2vec で動的に生成する。

- ⑤設問 2 の内容を入力する。

設問 2 は主制御関連の問題となっている、解答の内容は文章の並び替えを行う、内容は正答の順番に文章を入力する。

- ⑥設問 3 の内容を入力する。

設問 3 は機能部分の問題となっている、解答の内容は文章の並び替えを行う、正答は正答の順番に文章を入力する。

- ⑦設問 4 の内容を入力する。

設問 4 は機能の入出力を選択する問題となっている、機能に対して必要なデータと入出力されるデータを入力する

- ⑧設問 5 の内容を入力する。

設問 5 はシステム全体の流れと機能の内部の流れをチャート形式で構成する問題となっている。ピースを並び替えて、正しい順にシステムを構成する。

3.2 問題の格納と表示

問題構成に必要な文章やパラメータをデータベースに格納する。

297	6	0	1	システムにログインするときに一般的にIDの他に使われるものは?
298	6	1	2	パスワード
299	6	1	2	ログイン
300	6	1	2	ID
301	6	0	3	ウェブページの表示に変わるアプリケーションは?
302	6	1	4	ブラウザ
303	6	1	4	Internet Explorer
304	6	1	4	ウェブブラウザ

図 6 データベースの内容例

問題の表示は学習者が問題選択画面で選択した問題番号をもとにデータベースから取得・表示する。データベース内の情報から問題生成に必要な情報を取得したあと、ブラウザ側で問題を生成・表示する。

4. Word2vec 概要

4.1 Word2vec 概要

Word2vec とは単語をベクトル表現化して次元圧縮を行う定量化手法である[2]. 単語を数値として扱うことが可能になるため、単語同士の類似度や単語間の加算・減算を行うことが出来る[3].

4.2 ベクトル表現化の理論

Word2vec で単語に数値を持たせるために必要なものは次の 2 点である.

① わかち書きテキストデータ

単語毎にスペースで区切られたテキストデータ

② Word2vec 本体

Word2vec を用いてテキストデータ内にある単語群を解析し、それに対して指定数のベクトル (次元) を生成する. その後にベクトルと単語に対してどれだけの関連度数があるのかを計算する. 以下に例えを示す.

表 1 Word2vec によるベクトル演算の結果例

	単語 A	単語 B	単語 C	単語 D
次元 1	0.05	0.55	0.3	0.1
次元 2	0.95	0.15	0.7	0.2
次元 3	0.03	0.02	0.05	0.85
次元 4	0.54	0.3	0.2	0.5

わかち書きされたテキストデータを解析し頻出するキーワードを「次元」として扱い、その次元と「単語」に対しての類似度 (関連度) を数値として取得する. 1.00 に近いものほど関連度が強いものとして扱われる.

4.3 関連度計算

Word2vec の仕様として、「単語」の周辺にある「単語」は関連度のあるものとして扱う. 例として、わかち書きテキストの中身に以下の内容を考える.

「明日の 朝ごはん は パン と スープ が飲みたい」

Word2vec ではキーとなる単語に対して周囲のどの範囲の単語までを関連語として扱うかをパラメータとして指定出来る[4]. 例えば、「パン」と言うキーワード

に対して周囲 3 つまでを関連があるとした場合、結果は「パン」に対して「朝ごはん」と「スープ」が関連のあるキーワードとして出力される.

このパラメータの値を大きくすることで数値の精度としては、範囲を狭くした時に比べ落ちるが範囲の広い結果を取得することが出来る.

4.4 Word2vec の学習プロセス

わかち書きテキストを用意する為に、複数のウェブページからテキストを収集する必要がある. 今回は以下の図 7 の流れで Word2vec による学習を行った.

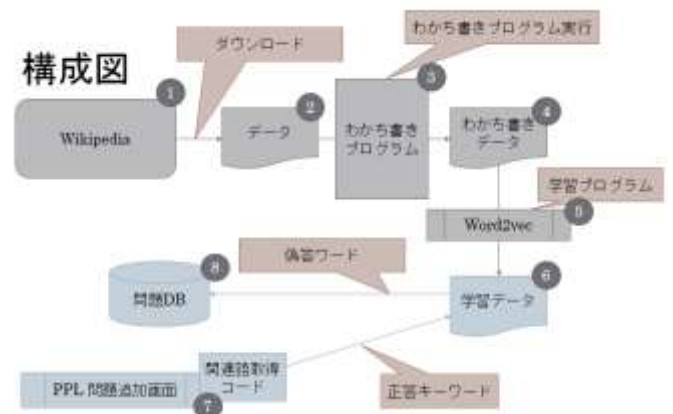


図 7 データを Word2vec で学習させる例

- ① Wikipedia から記事のダンプファイルをダウンロードする
- ② ダウンロードされたデータ (純粋なテキストファイル)
- ③ わかち書き用のプログラムでわかち書きテキストに変換
- ④ わかち書きテキストを出力
- ⑤ Word2vec によるわかち書きテキストの学習
- ⑥ 学習データの出力

5. PPL システムへの応用

5.1 偽答の動的生成

Word2vec により学習結果の入った学習データが出力された状態で、この学習データに対してプログラムから正答キーワードを与えることでそのキーワードに対して関連度の高い「単語群」が抽出される. PPL システムでは、この「単語群」の中から数値の高いもの上位 2 つを偽答キーワードとしてデータベースに正答キーワードと一緒に格納する.

5.2 問題格納プロセス

PPLシステムの偽答の生成と追加の流れを図8に示す。出題者は問題追加画面へアクセスし各設問の内容を入力し、格納実行ボタンを押す。

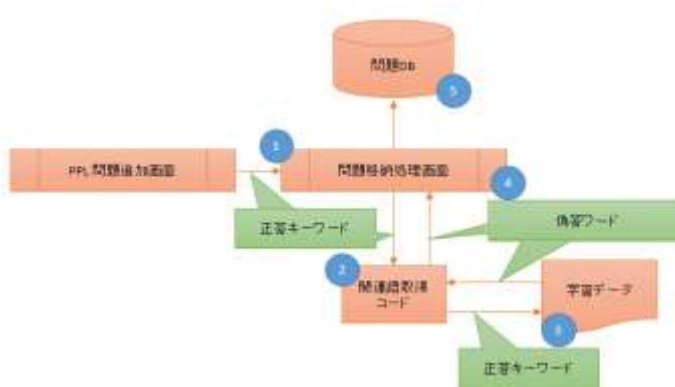


図 8 偽答の取得と格納の流れ

- ①問題格納処理のプログラム中で正答キーワードをキーワードとして別途で用意された関連語取得プログラムに渡す
- ②関連語取得プログラムが Word2vec に対してキーワードを与えて関連語を抽出する
- ③関連語として得られた単語群の上位 2 つを格納処理プログラムに返す
- ④偽答キーワードを問題格納処理プログラムから問題データベースに格納する

(b)操作部分:入力文章 = システムにログインするときに一般的にID他に使われるものは?
count_length_input = 1
(b)通常文: 格納完了
選択3 = : 3

```
Array  
{  
  [0] => パスワード  
}  
---偽答候補出力結果---  
array(2) [  
  [0] =>  
  string(12) "ログイン"  
  [1] =>  
  string(2) "ID"  
]  
出力結果mein_operation_data_select  
Array ( [0] => パスワード [1] => ログイン [2] => ID )  
(b)セレクトフォーム文: 格納完了
```

図 9 偽答の格納結果

6. おわりに

プログラミング学習に関する情報として、複数のウェブサービスやアプリケーションが存在しているが、どの内容においても学習者の理解度の違いや出題者側にかかる負担などの課題がある。ウェブページに個人

がアクセスして学習を行えるような教育システムは今後より高い需要が生まれると考えるが、需要の高まりと同時に学習者に対して学習効果の出る内容・環境を整備することは難しい。今回は学習者に段階的詳細化の重要性の理解と問題解決能力の向上を図るシステムを開発し、その中で出題者が効率的に問題を生成出来る機能を実装した。評価結果から、Word2vec を用いた現時点での偽答生成は、対義語などの表現に弱いなどの問題があることが分かった。

今後は、偽答の向上の為の事前単語フィルタリングや格納処理時の偽答候補の選択機能などの実装を行う予定である。

謝辞

本研究は、JSPS 研究費 15K01086 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,プログラミング教育,
http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1375607.htm (2017年2月1日確認)
- (2) DeepAge,ビッグデータ,Word2vec,
https://deepage.net/bigdata/machine_learning/2016/09/02/word2vec_power_of_word_vector.html#word2vec
とは,(2017年2月1日確認)
- (3) qiita,米 google の研究者が開発した Word2Vec で自然言語処理(独自データ),
<http://qiita.com/okappy/items/e16639178ba85edfee72>
(2017年2月1日確認)
- (4) qiita,word2vec のソースを読んでみた,
<http://qiita.com/mash0510/items/347964f3eb2e080ea7a4>,(2017年2月1日確認)