

# 多人数演習における多くの学習者が自信を持ってない箇所の検出・集計

鈴木康平・高瀬治彦・北英彦  
Email: 423m222@m.mie-u.ac.jp

三重大学大学院工学研究科電気電子工学専攻

◎Key Words 記述式解答, 入力過程, 自信がない箇所の検出

## 1. はじめに

多人数講義において、講師はより良い講義のために、短い時間で学習者の理解状況を把握する必要がある。しかし、多人数講義についての最大の問題は、学習者の理解度を講師が把握することにある<sup>(1)</sup>。

学習者の理解状況を把握する方法の一つに、講義中に簡単な課題を課して、机間巡視や学習者の解答などから、学習者の様子を確認する方法がある。この際、記述式の解答を要求することで、細かく学習者の理解状況が把握できる。その反面、多人数講義においては、解答時間中に学習者全員の解答を読むことは難しい。そのため解答終了後すぐには、学習者の理解状況に応じた指導ができない。

現在、さまざまな大学での PC の必携化<sup>(2)</sup>などが浸透し、演習も計算機を利用して解答の作成・提出を行う場面が増えている。このような状況のもとで、ICT 技術を利用したさまざまな演習支援が行われている。例えばニューラルネットワークを用いた文章分類により解答の正解/不正解を分類する試みがある<sup>(3)</sup>。この手法では、提出された解答を判定することで、理解状況把握の支援をしている。しかし、解答を解析する手法では、解答が完成している必要があり、解答が提出されるまで解析結果を得ることができない。また解答に頼らず、学習者のふるまいから理解状況に関する情報を得る手法として、学習者の集中度をカメラで撮影した画像から推定する試みがある<sup>(4)</sup>。しかし、学習者のふるまいから解析する手法では、解答の「どこ」の部分につまづいているのか把握することはできない。さらに、学習者のふるまいと解答内容を合わせ分析する方法として、計算問題における解答の習熟度が低い箇所をペンストロークの停滞箇所から検出する手法<sup>(5)</sup>や学習者の自信がない箇所を解答の入力停滞から検出する手法<sup>(6)</sup>がある。前者は、数字や文字などある程度解答パターンが決められた問題でしか、停滞した箇所を検出することはできない。後者は、各個人の状況が判別できるのみであるため、多人数で実施する演習において、全体の状況を把握するには向いていない。

本研究では、多人数で実施する演習において、学習者のふるまいと解答内容を合わせることで、多くの学習者が自信を持ってない箇所を検出し、講師に提供することを試みる。本稿では、文献(6)の手法により各学習者の自信がない箇所を検出した後、全学習者の自信がない部分に関連した記述をその内容に従い分類し講師に提供する。

## 2. 解答の入力停滞から学習者の自信がない箇所を検出する手法

この節では、各学習者が自信を持ってない箇所を講師に提供するために、入力過程における入力の遅れから学習者の自信がない箇所を逐次検出する方法として、文献(6)の方法について説明する。

この方法は、入力された解答だけでなく、各文字がいつ入力されたかという情報(入力過程)を用いて、解答のどの部分に学習者が自信がないのかを検出する。解答の入力が停滞する箇所の近辺に自信がない箇所が存在する<sup>(7)</sup>ことに着目し、入力の停滞を検出する。これは、これまでの解答入力速度と、日本語変換の際の入力開始から確定までの入力速度を比較することで行う。この手法により、変換操作毎に、学習者が解答を提出する前から、自信がない箇所の検出ができる。

表 1 はある学習者が「右ねじの法則を説明せよ」という問題に対して「ある電流を流すことにより、…」と解答を入力した解答の一部を示したものである。この学習者は「…/電流を/流すことに/より/…」と/の位置で変換を行った。それぞれの区間において入力が遅延したかどうかの判定を行う。表 1 には、解答開始時点からその時点までの区間の平均入力速度と変換した区間の入力速度(区間平均速度)、それぞれの区間の遅延の有無を判定した結果を示す。遅延の有無は平均入力速度と区間平均速度を比較し、区間平均速度が平均入力速度を下回った場合は、入力が遅延・停滞したと判定する。表には、遅延の有無を記した。このように、変換確定毎に平均入力速度と区間平均速度を求めることで、遅延した箇所の判断をするため、変換確定毎に逐次判定する。これをもとに、取得した箇所の前後 2 区間を自信がない箇所として抽出している。

表 1 ある学習者の解答の入力過程の一部の遅延状況

	…	電流を	流すことに	より	…
平均入力速度 (字/秒)		0.65	0.57	0.79	
区間平均速度 (字/秒)		0.56	0.49	2.1	
遅延の有無		有	有	無	

この手法の課題は 2 つある。第 1 の課題は、検出結果が、読みにくいことである。この手法では入力が遅延・停滞した箇所は、変換単位で抽出される。そのため、

必ずしも意味の切れ目で区切られているわけではなく、読みにくくなる場合が生じる。第 2 の課題は、多人数の動向がわかりにくいことである。この手法では各個人に対して検出を行うため、多人数の場合は、検出結果が多数得られる。多数の解答の断片を並べて提供されても、全体の動向を把握することは困難である。

### 3. 提案法

本節では、前節で述べた2つの課題(抽出結果が読みにくい、多人数の動向がわかりにくい)を解決する方法について検討する。ここでは、意味をとらえやすいように抽出する手法と、抽出結果を要約する手法についてそれぞれ検討する。

まず、意味をとらえやすいように抽出する手法について検討する。ここでは、抽出結果を意味をとらえやすいよう長さに調整することで、抽出結果を分かりやすくする。文章において、意味を持つ最小の単位は単語である。そこで、第 1 の抽出法として単語一つだけにする方法を考える。この手法では、元の抽出結果に含まれる単語を、一つずつ抽出する。しかし、抽出した各単語それだけでは、「どこ」で入力停滞したのかを把握するには適しているが、「どのような内容」を記述する際に入力停滞したのか把握しにくい。そこで、抽出範囲をより長くすることを試みる。第 2 の抽出方法として、文節単位での抽出することを考える。ただし、単一文節では第 1 の方法と同じく、抽出結果が短く、内容を把握しにくいだらう。そこで、元の抽出範囲に含まれる文節と、それ係る文節を抽出する。ただし、これでも十分に意味が取れない場合もあるだろう。そこで第 3 の方法として、関係のある語を一通り抽出する方法を考える。この方法では、元の抽出結果に含まれる各語について、その語に関係している語を連鎖的に抽出する方法を考える。具体的には各語の依存関係に注目する。具体的な手順を以下に示す。まず、元の抽出結果に含まれる各語を抽出する。次に、元の抽出結果に含まれる各語について、語の依存関係による直接の係り受けの関係にある語を抽出する。さらに、一つ前の手順で抽出された各語について、そこに係る語をすべて抽出する。これを新たに抽出する語がなくなるまで繰り返す。最後に、ここまでの操作で抽出されたすべての語を、原文の順に連結したものを抽出結果とする。この手法では、下の抽出結果に関係する語がすべて抽出されるため意味は取りやすくなる。その反面、抽出結果が長くなる。

以上をまとめると、抽出単位の候補としては、単語・文節・関連語が挙げられる。これらの違いについては実験の節で比較する。

本稿では、文節・語の係り受けに着目した切り出しには、係り受け解析器と呼ばれる自然言語処理のツールを用いる。係り受け解析とは、文の構成要素(文節・語)間の依存関係を明らかにする方法で、どれがどこに係っているかを解析する方法である。よく使われる係り受け解析器としては語間の関係性を出力する GiNZA<sup>(8)</sup>、文節間の関係性を出力する南瓜<sup>(9)</sup>などがある。

次に、抽出結果を要約する手法について検討する。多数の抽出結果をそのまま提供することは、概要を素早く把握するには向かない。そこで、抽出結果をクラ

スタリング(分類)し、各クラスタの代表と大きさを提供することで、多量の抽出結果をすべて閲覧しなくても良いようにする。ここでクラスタリングとは、与えられた多数のデータをその類似度に応じてグループ化する手法である。主要なクラスタリング手法では、各データがベクトルとして表されていることが必要である。文章などをベクトル化する手法としては、使用単語の頻度に基づく Bag of Words 法<sup>(10)</sup>や、言語モデルにより意味をベクトル化する埋め込みベクトル法などがある。ここで扱う解答は、学生の自由記述によるものを想定しているため、使用単語・表現にはゆらぎが含まれる。そのため、単一単語より長いもののクラスタリングを行う際は、後者の利用を検討する。

埋め込みベクトルの生成には、単語から埋め込みベクトルを生成する word2vec<sup>(10)</sup>、文章から埋め込みベクトルを生成する doc2vec<sup>(10)</sup>、任意の長さのフレーズから埋め込みベクトルを生成する Universal Sentence Encoder (USE)<sup>(11)</sup> などが有名である。

クラスタリング手法には、階層型クラスタリングや K-means 法などがある<sup>(12)</sup>。本稿では各クラスタの代表を提供するため、K-means 法を用いる。また、各クラスタの代表は、K-means 法の各クラスタの代表点に最も近い抽出結果を選択する。

### 4. 実験

この章では、提案法により実際の授業の演習問題の解答に対して、前章で検討した手法を適用し、その抽出結果を確認する。

#### 4.1 実験条件

本研究の実験では、Web システムをつうじて演習の解答を行う。使用したシステムは、学習者が打鍵したキーの打鍵時刻を一字ごとに収集できる。各学習者は簡単な記述式演習問題を解いた。学習者は、本学の電気電子工学科の講義「計算機工学」の受講者 40 名であり、演習問題は計算機のハードウェアに関する質問である。具体的な演習問題と解答例は以下のとおりである。

問：ハーバードアーキテクチャのメリットとデメリットを説明せよ

解答例：メリットはフォン・ノイマンのボトルネックを回避でき、高速な動作を実現できる。プログラムとデータのビット幅を揃える必要がない。デメリットは回路が複雑になることである。

本稿では、形態素解析と係り受け解析は GiNZA、埋め込みベクトルは USE、クラスタリングは K-means 法を使用する。

#### 4.2 実験結果

表 2, 3, 4 に、各種法で抽出・クラスタリングした結果の上位のクラスタの代表とクラスタの大きさを示す。表 2 は、抽出した各語について、その語を使用した解答者数を示している。なお結果は、使用者数順にした上位 9 語分のみを記した。表 3 は文節単位で抽出した結果を USE で埋め込みベクトル化してクラスタリングした結果に

ついて、クラスタサイズの上位 9 件分を記した。表 4 は、関連語で抽出した結果を USE で埋め込みベクトル化してクラスタリングした結果について、クラスタサイズの上位 9 件分を記した。なお、これらの表に記した件数は、各停滞箇所に対して別個に処理した結果を計数したものであり、重複して計数しているものも含まれる。

また表 5, 6 にクラスタリングの結果、同じクラスタに分類された抽出結果とその数を示す。表 5 は表 3 における最初のクラスタ「データのビット幅を」と同じクラスタに属しているものすべてを示している。また、表 6 は表 4 における最初のクラスタ「プログラムメモリとデータメモリが」と同じクラスタに属しているもののうち頻度が 2 以上のものを示している。なお、各表に記した数は、文字列が完全に一致したものの数を示している。

表 2 自信がない箇所の頻出単語と人数

自信がない箇所の頻出単語	人数
デメリット	20
プログラム	18
複雑	17
フォン・ノイマン	15
ボトルネック	15
回避	14
回路	14
メリット	13
メモリ	13

表 3 文節単位の代表とクラスタサイズ

文節単位の代表	サイズ
データのビット幅を	22
デメリットとしてはなる	22
データメモリが分離している	20
プログラムとデータの	18
データ用のバスが	17
複雑になるという	16
動作が実現できる	15
プログラムメモリとデータメモリが	15
回路がなりがちという	15

表 4 関連語の代表とクラスタサイズ

関連語の代表	サイズ
プログラムメモリとデータメモリが	48
プログラムメモリとデータメモリが分離しているため、フォン・ノイマンのボトルネックを回避でき、プログラムとデータのビット幅を揃える必要がない	38
デメリット:回路が複雑になりがちだという	
フォン・ノイマンのボトルネックを回避して	34
回路が複雑になってしまうという	29
ハーバードアーキテクチャのメリットは命令用とデータ用のバスが分離しているので、ノイマン型のボトルネックを回避して高速な動作が実現できる	23
プログラムとデータのビット幅を	21
ハードウェア構成が複雑になってしまう	17
プログラムとデータの	16
命令用とデータ用の	16

表 5 表 3 の最初のクラスタの内容

抽出結果	数
データのビット幅を	12
ビット幅を揃える	3
ビット幅をそろえる	2
データのビットを	1
データの bit 幅を	1
ビット幅をそろえることができる。	1
ビット幅を揃えなくてもよいという	1
プログラムの幅が	1

表 6 表 4 の最初のクラスタの内容

抽出結果	数
プログラムメモリとデータメモリが	8
プログラムメモリとデータメモリの	3
プログラムメモリと	3
命令メモリとデータメモリが	2
メモリ・データメモリと	2
プログラムメモリとデータメモリが分離している	2
デメリットとして、命令側のメモリが余っていたとしても	2
CPU からのメモリがプログラムとデータそれぞれ分断されている	2

## 5. 考察

表 2 では、抽出結果が読みやすく、解答の「どこ」の部分につまずいているのか把握しやすいだろう。表 2, 表 3, 表 4 より抽出単位を単語・文節・関連語のように広げると、意味を捉えやすくなると感じる。そのため、「どのような内容」があったのかが把握しやすくなっている。ただし、表 4 の第 2 位の項目のように長過ぎると、読みづらくなる点にも注意が必要だろう。実際に使用する際に

は、抽出単位を変化できるようにすると良いだろう。

表 5, 6 より、埋め込みベクトルに基づいてクラスタリングした結果について検討する。表 5 では、全 22 個の抽出結果が、8 種類の抽出結果を含んでいる。「プログラムの幅が」といった、異なる内容が含まれているが、概ね意味が類似したものでクラスタが構成されている。表 6 では、全 48 個の抽出結果が、頻度 1 のものを含めて 32 種類の抽出結果を含んでいる。長い抽出結果の内容は異なるものが多いが、内容的には短いものの内容を含んだものが分類されている。また、いずれの分類結果も文字列としては大きく異なるものが同じクラスタに分類されていることも分かる。このように、抽出長さの調整は必要だが、埋め込みベクトルでベクトル化することにより、意味に基づいた分類ができると言えるだろう。

## 6. おわりに

本稿では、解答の入力過程から学習者の自信がない箇所を検出した後、多人数で実施する演習において、多くの学習者が自信を持っていない箇所を講師に提供することを目的とした。このために、文の構成要素間の係り受けの関係に着目し、単語・文節・関連語それぞれを抽出単位として提供することを提案した。簡単な実験により、抽出単位を単語・文節・関連語のように広げると、意味を捉えやすくなることを確認した。加えて、文節・関連語の際は、埋め込みベクトルでベクトル化することにより、意味に基づいた分類ができることを確認した。しかし、関連語では抽出結果が長すぎる場合もあるため、今後は関連語を抽出する際の抽出範囲についても検討する。

## 参考文献

- (1) 西澤泰彦:“多人数講義における問題点と教育方法”, 名古屋高等教育研究, 第 6 号, pp.52-56(2006)
- (2) 宇治橋祐之, 渡辺誓司:“GIGA スクール構想と「オンライン学習」に向けたメディア利用～2020 年度「新型コロナ下の小学校, 中学校, 特別支援学校でのメディア利用に関する調査」から～”, メディア研究部, pp.50-58(2021)
- (3) 寺田凜太郎, 久保顕大, 柴田知秀, 黒橋禎夫, 大久保智哉:“ニューラルネットワークを用いた記述式問題の自動採点”, 言語処理学会発表論文集, 第 22 回年次大会, pp.370-373(2016)
- (4) 藤白智也, 浅利恭美, 宮田真宏, 山田徹志, 大森隆司:“授業中の画像からの姿勢推定による集中度の評価”, 人工知能学会全国大会論文集, 第 34 回全国大会, 一般社団法人人工知能学会(2020)
- (5) 飯山将晃, 中塚智尋, 森村吉貴, 橋本敦史, 村上正行, 美濃導彦:“ペンストロークの時間間隔を用いた解答停滞箇所の検出”, 情報システム情報学会誌, Vol. 34, No. 2, pp. 161-171(2017)
- (6) 鈴木康平, 高瀬治彦, 北英彦:“記述式演習からの学習者の自信がない箇所の入力過程を用いた逐次検出に関する一検討”, 2023 PC Conference 論文集, pp.237-239(2023)
- (7) 鈴木公貴, 木村聖, 高瀬治彦, 北英彦, 川中普晴:“解答の入力停滞からの学生の自信がない語の検出の試みー停滞の判定基準に関する検討ー”, 2018PC Conference 論文集, pp.48-51(2018)
- (8) 松田寛:“GINZA - Universal Dependencies による実用的日本語解析”, 学会記事, 2020 年 27 卷 3 号, pp.695-701(2020)
- (9) 工藤拓, 松本裕治:“チャンキングの段階適用による日本語係り受け解析”, 情報処理学会論文誌, Vol.43, No.6, pp.1834-1842(2002)
- (10) 山本和英, (2021), テキスト処理の要素技術, 日本:近代科学者
- (11) Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, Ray Kurzweil: “Universal Sentence Encoder”, arXiv:1803.11175, (2018)
- (12) 新納浩幸, (2007), R で学ぶクラスタ解析, 日本:オーム社