

言語学習を対象とした自学自習システムの研究 ～システム開発に関する研究成果～

梅澤 克之^{1,a)} 中澤 真^{2,b)} 小林 学^{3,c)} 石井 雄隆^{4,d)} 中野 美知子^{5,e)} 平澤 茂一^{6,f)}

概要：本研究の最終的な目的は、言語学習を統一的な枠組みで捉え、相手（学習者）を感じて助言を行う人工教師を搭載した自学自習システムを開発し、その評価を行うことである。「相手を感じる」とは、学習者の学習状況をシステム側が把握することを指す。本報告では、上記最終的な目標に対する途中経過を示す。具体的には自学自習システムに必要な「ケアレスミス」、「文法誤り」、「論理誤り」を検出する方法に関する研究成果および、「相手を感じる」ための生体情報としての脳波を効率的に収集するために開発した脳波計測システムについて述べる。

A study on Language-Learning Self-Study System —Research Results on System Development—

1. まえがき

大規模公開オンライン講座などの普及に伴い、学習者が自分のペースで学習を進める非同期型遠隔教育が普及している（大学 ICT 推進協議会）[1]。このように遠隔による自学自習の需要は大きくなっている。教室における対面型の講義においても学生の主体性を伸ばす方法論として反転授業が注目されており、対面授業に先立って行う自学自習が重要な役割を果たす [2]。

また、2016 年には「学習指導要領 [3][4]」でグローバル化と教育の国際化に向けて英語教育とプログラミング教育

の早期導入が強調された [5]。プログラミング教育に関しては、中学校で必修となり、政府の成長戦略の中では「義務教育段階からのプログラミング教育等の推進」が盛り込まれた。また文部科学省が小学校でのプログラミング教育の必修化を発表した。さらに総務省も 2025 年までに IT 人材を新たに 100 万人育成する方針を示し、プログラミング教育を推進している。

このように英語教育とプログラミング教育は共に重要であるが、これら相異なる言語学習における e ラーニング教材や履歴データ分析は独立に考えられてきた。プログラミング言語は合理的に人工的に作られているのに対し、自然言語である英語には例外がある。また、両者とも「読む」と「書く」があり（英語は「聞く」「話す」もあるが）、両者とも書く際には論理的思考が必要である。また、「読む」に関しては多くの文献やメディアで読解力の低下が指摘されている。このように英語とプログラミング言語は自然言語と人工言語なので異なる部分もあるが、学習過程において共通的な部分も多いと考えられる。しかし、読解問題や記述式問題の解答過程に着目して英語とプログラミング言語を統一的に分析し、それらの関係性を明らかにし、お互いに補完し合うような教育システムを組むことで多大な相乗効果を目指した研究はいまだかつてなされていない。上記を達成するため、今後の我が国における重要科目である英語とプログラミング言語に対し、個人毎の学習状況に応

¹ 湘南工科大学 工学部
Shonan Institute of Technology, Kanagawa, 251-8511, Japan

² 会津大学 短期学部
Junior College of Aizu, Fukushima 965-0003, Japan

³ 早稲田大学 データ科学総合研究教育センター
Waseda University, Tokyo, 162-0042, Japan

⁴ 千葉大学 教育学部
Chiba University, Chiba, 263-8522, Japan

⁵ 早稲田大学 教育・総合科学学術院
Waseda University, Tokyo, 169-8050, Japan

⁶ 早稲田大学 理工学術院
Waseda University, Tokyo 169-8555, Japan

a) umezawa@info.shonan-it.ac.jp

b) nakazawa@jc.u-aizu.ac.jp

c) mkoba@waseda.jp

d) yishii@chiba-u.jp

e) nakanom@waseda.jp

f) hira@waseda.jp

じた自学自習が可能のようにアドバイスを行う人工教師と両者に共通なプラットフォームを構築することが研究対象である。

本研究では、学習者の生体情報や学習履歴・学習記録を取集・分析し、分析結果をシステム側が自動的に学習者にフィードバックするエージェントを人工教師と呼ぶ。本研究では学習者を観察・分析し個々の学習者に適した学習コンテンツの提示や適切な指示を行う人工教師を搭載した自学自習システムを開発する。

2. 従来研究

2.1 学習履歴の蓄積システム

学習者がプログラムを完成させて行く過程や、その過程でどのような課題に直面し、どのように解決したかを知るのは難しく、それらを理解するためにはバージョン単位の履歴では不十分であり、どのように変更されてきたのかユーザの操作単位で詳細に記録を取るものの有効性が指摘されている [6][7]。文献 [6] では、蓄積した学習履歴の活用として、操作履歴を表示する機能、表示機能に対して表示条件を設定してフィルタリングする機能、過去の任意の状態のソースコードを表示・復元する機能を提案している。文献 [7] では、詳細な学習履歴をリアルタイムに蓄積し分析することで、該当の学習者への良い点、悪い点のフィードバックを可能にし、教師側は学習者の躓きの箇所がわかる機能を有しているが、蓄積した学習者全員の履歴を自動で分析したり、分析結果をもとに品質を向上させるまでには至っていない。

2.2 脳波計測の学習への応用

脳波の波形を関連事象とともに観測すると精神状態の指標として用いることができるということは従来から心理学や脳科学の研究で経験的にわかっている [8]。また、人間の精神状態を観測するために、得られた脳波に対して離散フーリエ変換を施して得られた α 波や β 波を用いた研究がなされている [9]。また、 β 波は思考状態と関連性が高いとして、Giannitrapani らは知的作業と脳波の関係を調査し、知能テストを受けている最中の健常者の脳波を測定した [10]。その結果、 β 波の低周波成分（低 β 波）が読解テスト、数学テスト、図形整列テスト中に優勢となることを示し、 β 波が思考状態を推定する指標としてある程度有効であることを示した。さらに、人間の思考状態を観測するために α 波や β 波のパワースペクトルや、 α 波や β 波の脳波全体に対する割合、あるいは、 α 波と β 波の比率を測ることが有効とされている [11]。このように従来研究により脳波を計測することによって学習者の学習状況を把握することの可能性が示されている。本研究でもこのような従来研究に従い、まずは脳波（代替となる生体情報の探求も研究の一部である）を計測することにより「相手を感じる」

ことを実現する。

3. 本研究の全体概要

本研究では、学習者の生体情報や学習履歴を取得したうえで、教師によるリアルタイムの対応が不可能な授業時間外でも個々の学生の学習状況に適する学習方法を提示できる自学自習システムを開発する。開発したシステムを用いて実証実験および評価を行う [12]。本研究の全体概要を図 1 に示す。

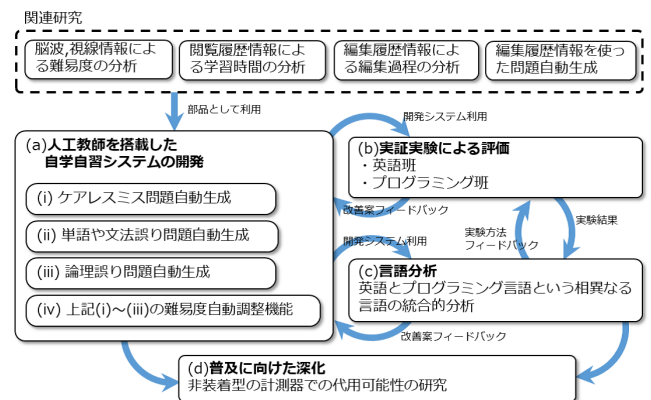


図 1 本研究の全体概要

本研究は下記の 4 つの研究課題に大別できる。

- (a) 人工教師を搭載した自学自習システムの開発
- (b) 英語およびプログラミング言語を対象とした実証実験による評価
- (c) 英語とプログラミング言語という相異なる言語の学習履歴の統合的分析
- (d) 普及に向けての非装着型の計測器での代用可能性追求

ここで前述の (a) では、(i) ケアレスミス（脳波情報と解答時間から判定し、ケアレスミスをし易い問題を集中的に出題する機能）、(ii) 単語（英語における単語やプログラミング言語における予約語）の綴りや文法（構文）の誤りが多い学習者に対しては単語や文法の理解を向上させる（単語の綴り間違いや文法誤りを修正させる）問題を出題する機能、(iii) 文法的には正しいが論理的な誤りを含む問題を訂正させる問題を出題する機能、また、(iv) 上記 (i)~(iii) について脳波や視線などの生体情報や学習履歴情報を組み合わせて学習に集中できていない状態、学習内容が簡単すぎる状態、学習内容が難しすぎて理解できていない状態、部分的に理解できない箇所がある状態を判別し、リアルタイムで課題の難易度を調整する機能を持った自学自習システムを開発する。(i)(ii)(iii) によりケアレスミスを起こしやすい問題、文法エラーを起こしやすい問題、論理エラーを起こしやすい問題を抽出できるので、これらの機能と (iv) を組み合わせることにより、学習者毎に最適化された難易度の問題を出題することが可能となる。

また、(b) では開発したシステムを用いて英語およびプ

プログラミング言語という観点で実証実験により評価を行う。(c)では実証実験の結果を受けて個々の学習者の英語の学習過程とプログラミング言語の学習過程に何等かの相関があるのか、双方の教育に活用することで相乗効果が見込めるか等の分析を行う。(d)では研究成果の普及のために、脳波計での判定結果と同様の結果を示す非装着型の計測器(例えば Web カメラでのまばたきの計測など)での代用可能性を追求する。

4. 自学自習システムの最終形

本研究の研究課題「(a) 人工教師を搭載した自学自習システムの開発」で目指す最終的な自学自習システムを図2に示す。このシステムはケアレスミスや文法誤り、論理的な誤りなどを含みそれらを訂正させる問題を出題する「出題サーバ」と、回答を受け付ける「回答サーバ」と、学習時の生体情報を収集する「学習者を感じるサーバ」を含む。また、脳波や視線などの生体情報や学習履歴情報を組み合わせて学習に集中できていない状態、学習内容が簡単すぎる状態、学習内容が難しすぎて理解できていない状態などを判別し、リアルタイムで課題の難易度を調整する「分析サーバ」を持つ。分析サーバでは、ケアレスミスや文法(構文)の誤り、論理的な誤りなどを検出する機能と、脳波や視線などの生体情報や学習履歴情報を組み合わせて学習に集中できていない状態、学習内容が簡単すぎる状態、学習内容が難しすぎて理解できていない状態、部分的に理解できない箇所がある状態を判別する機能を持つ。

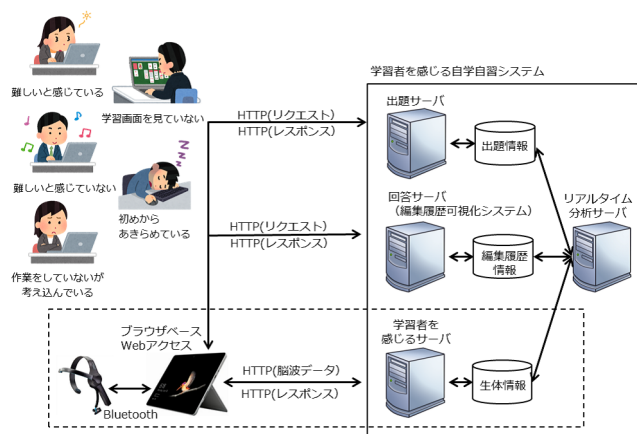


図2 自学自習システムの全体像

5. 研究成果

本章では、前章で示した4つの研究課題のうち「(a) 人工教師を搭載した自学自習システムの開発」に関する研究成果を述べる。本研究の最終的な目的はプログラミング言語と英語を統合的に分析することであるが、本報告ではプログラミング言語のみを対象としている点にご注意頂きたい。なお、以下の5.1節～5.4節で述べる研究成果の詳細は

それぞれ個別の学会・研究会等で発表済みである。

5.1 ケアレスミスの判定

本節では、Java 言語学習中の回答時間と脳波の関係からケアレスミスを判定する方法についての研究[13][14]について述べる。

5.1.1 概要

本研究では、湘南工科大学の学部4年生3名を対象とし、プログラミングの学習時の脳波情報の計測実験を行った。課題の回答時間と脳波の関係に着目し、注意力が欠けた際のケアレスミスを検出することを試みた。これにより学習者が起こしやすいケアレスミスを把握し注意喚起を行ったり、ケアレスミスを起こしやすい問題を集中的に訓練したりすることでプログラミングの技術向上につなげることができる。と考える。

5.1.2 実験方法

学習内容はJava言語の基礎であり、7つの課題「1. 変数、演算」「2. 分岐」「3. 繰り返し」「4. 配列」「5. メソッド」「6. クラス1」「7. クラス2」を学習する。各課題にはその内容を解説する資料と理解度を測定するための10問の設問で構成される。まず、実験参加者は1つ目の課題の「1. 変数、演算」についての解説資料を読んで学習する。その際の脳波を5分間計測し、5分経過後に計測を一旦止める。引き続き「1. 変数、演算」についての設問を1問目から10問目まで順番に回答する。その際に脳波を計測する。各設問で計測を止めずに連続して計測する。計測時間がログとして残るので各設問ごとの脳波を切り出すことが可能である。これを課題「1. 変数、演算」から課題「7. クラス2」まで繰り返す。

5.1.3 ケアレスミスの判定方法の提案

今回我々は、ケアレスミスを「時間をかけず回答して誤答した」かつ「よく考えずに回答して誤答した」、つまり「適当に答えたために誤答した」と定義する。そのうえで、ケアレスミスの判定方法を提案する。まず、実験参加者毎の各設問の解答時間の平均値を求める。そして誤答した設問に着目し、さらに各設問の解答時間が平均値よりも小さいものに着目する。つまりこの段階では、時間をかけずに誤答してしまった設問について着目していることになる。つぎに、対象の設問を解いている際に1秒間隔で取得した脳波(β/α)の平均値と、1問目から対象の設問の一つ前の設問(対象の設問は含まない)の平均値(以降累積平均と呼ぶ)を比較し、累積平均よりも小さい値を抽出する。ただし、設問1に関しては、比較する累積平均値が存在しないので対象外とする。最後に、抽出した平均値とそれに対応する累積平均値に対して、 F 検定(等分散の検定)および t 検定(平均値の差の検定)を実施し、平均値に有意な差が認められるかを検定する。今回の提案では、将来的にシステムに組み込んでリアルタイムでケアレスミスの判

定を行うことを想定して累積平均と比較を行う提案を行った。学習後にバッチ処理できる場合は累積平均ではなく全体の平均との比較を行うことで先頭の設問の判定も行うことができる。

5.1.4 実験結果

実験参加者 1 はそもそも誤答が少なく実験で行った Java プログラミングの内容は得意だったと考えられる。実験参加者 2 に関しては、誤答の数は少なくはないがケアレスミスと判定されるものではなく、熟考したうえで間違えてしまったと考えられる。図 3 に実験参加者 3 の 6 章の各設問回答時の平均脳波を示す。丸で示した箇所が β/α が下がっていきそうな箇所である。この箇所が誤答で、かつ回答時間が短くて、かつ統計的に有意に β/α が低ければ、そこがケアレスミスという判定になる。この例では設問 7 のみがケアレスミスと判定された。実験参加者 3 の全課題に関しては、多くの誤答があり、課題 2 の設問 9、課題 4 の設問 9、課題 5 の設問 7、課題 6 の設問 7、課題 8 の設問 3 と 9 に関してはケアレスミスであると判定できた。実験参加者 3 は各課題について後半の設問になるとケアレスミスが目立つ結果となった。

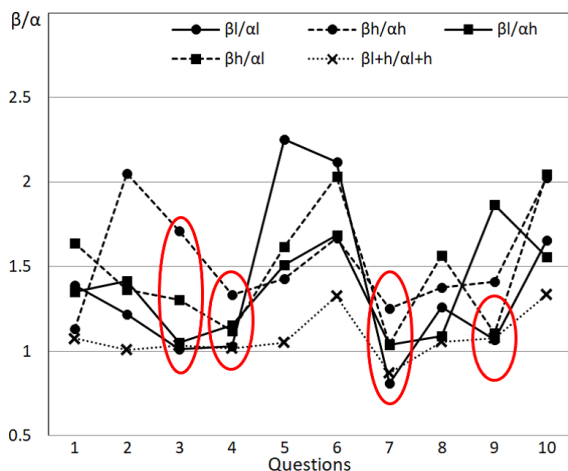


図 3 実験参加者 3 の 6 章の各設問回答時の平均脳波

5.2 文法誤りの検出

本節では、プログラミングの作成過程の編集履歴データから文法誤りを含むソースコードを抽出する研究 [15] [16] について述べる。

5.2.1 概要

湘南工科大学の約 90 名の学生が 16 週間のプログラミングの授業を受けた際の膨大な学習ログが蓄積された。これらの学習ログには、プログラムが完成するまでに修正されていく過程のソースコードをすべて含まれている。これらの情報を元にあえて文法誤りを含むソースコードを学習者に与えて、そこに含まれている間違いを修正するデバッグ練習用の問題を自動生成するデバッグ練習用の問題を自動

生成するデバッグ練習問題抽出システムを開発した。

5.2.2 システムの全体構成

今回開発したデバッグ練習問題抽出システムの全体構成を図 4 に示す。図 4 に示す通り、学習履歴を蓄積するまでは既存システムである編集履歴可視化システムを活用する。デバッグ練習問題抽出ツールは、編集履歴可視化ツールが蓄積した学習履歴を参照し、正解のソースコードと正解に至るまでの誤りを含むソースコードを比較し、誤りを含むソースコードを抽出する。抽出された誤りを含むソースコードは、編集履歴可視化システムの改造版で参照及び配布される。

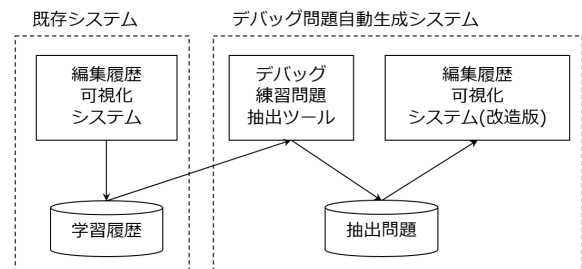


図 4 デバッグ練習問題抽出システムの全体構成

5.2.3 デバッグ練習問題抽出ツール

デバッグ練習問題抽出ツールは、編集履歴可視化ツールが蓄積した学習履歴、具体的には、各問題ごとの完成フォルダ (last) に存在する MyClass.java と、作成途中の MyClass.java を比較し、修正箇所数および各修正箇所の修正文字数を抽出する。抽出した「修正箇所数」および「最大修正文字数」をもとにフォルダを再構成する。なお、ソースコードの差分には、エラーを修正したための差分と、エラーを伴わない差分があるため、エラーを修正した場合のソースコードのみ抽出する事としている。開発したデバッグ練習問題抽出ツールを図 5 に示す。図 5 に示すように、何個までの修正箇所のものを抽出するか、何文字までの修正文字数のものを抽出するか、どの授業の問題を抽出するか等を指定出来るようにしている。なお、編集履歴可視化システムが蓄積する学習履歴のフォルダ構成を図 6 に、デバッグ練習問題抽出ツールが出力するフォルダ構成を図 7 に示す。

5.2.4 抽出アルゴリズム

本節では抽出アルゴリズムを示す。

- 1) 編集履歴可視化システムの全履歴 (図 6 の全データ) について下記の 2)~6) を繰り返す。
- 2) last.info に “end” と記載されているかを確認する (last.info は完了ステータスファイル)。
- 3) last フォルダ以外のフォルダについて、stdout ファイルに “エラー” が記載されているかどうかを確認する。
- 4) last フォルダ以外にエラーが記載されているフォルダ

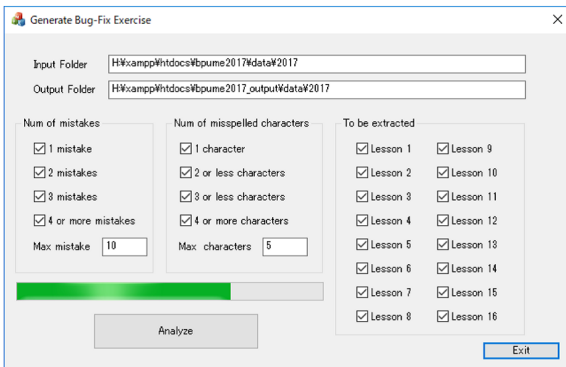


図 5 デバッグ練習問題抽出ツール

```

+ 0001 (レッスン番号)
+ 0002 (レッスン番号)
+ 17Axxx1(学籍番号)
+ 17Axxx2(学籍番号)
+ q001(問題番号)
+ q002(問題番号)
+ 20171006_092309(履歴)
+ 20171006_092508(履歴)
- MyClass.class
- MyClass.java
- stdin
- stdout
+ last (最終履歴)
- MyClass.class
- MyClass.java
- stdin
- stdout
- last.info(完了ステータス)

```

図 6 編集履歴可視化システムのフォルダ構成

```

+ 0001 (レッスン番号)
+ 0002 (レッスン番号)
+ q001(問題番号)
+ q002(問題番号)
+ 01-01-000001(個所数-文字数-連番)
+ 01-01-000002(個所数-文字数-連番)
+ 01-01-000003(個所数-文字数-連番)
+ before
- MyClass.java
+ last
- MyClass.java
+ 02-01-000004(個所数-文字数-連番)
+ before
- MyClass.java
+ last
- MyClass.java

```

図 7 デバッグ練習問題抽出ツールが出力するフォルダ構成

の MyClass.java と last フォルダの MyClass.java の差分をとる。

- 5) その際に、差分の箇所数と 1 つの差分に最大何文字含まれているかをカウントする。
- 6) 差分の箇所数と 1 つの差分に最大何文字含まれているかによって適切なフォルダ構成に従って MyClass.java ファイルをコピーする。

上記アルゴリズム中の差分の計算は、UNIX の DIFF コ

マンドで用いられている 2 つの要素列の最長共通部分列 (LCS: Longest Common Subsequence) とある要素列を他の要素列に変換するための最短手順 (SES: Shortest Edit Script) に基づく手法 [17] を用いた。

5.3 論理誤りの分析

本節では、プログラミングの作成過程の編集履歴データからコンパイラがエラー情報を出力しない論理誤りを含むソースコードを抽出する研究 [18][19] について述べる。

5.3.1 概要

前節に示したデバッグ練習問題抽出システムは文法エラーを対象にしている。これに対して本研究では、同じ学習履歴を利用して、論理エラーの分析を行った。論理エラーは文法的には誤りはないためコンパイラはエラー情報を出力しない。よって機械的に検出することは難しい。本研究では、大量のプログラミング学習履歴を蓄積し分析することによって間違いを起こしやすい論理エラーを自動で検出することができた。この結果を用いることで論理エラーを修正するデバッグ練習を行うことができる。

5.3.2 データ収集を行った授業について

我々は、反転授業の自習時の e-ラーニングの学習ログを取得し、自習時の理解度が高い学生のグループ、自習に時間をかけなかったために理解度が低い学生のグループ、自習に時間をかけたが理解度が低い学生のグループに分けて教場での対面授業を行う「グループ分け反転授業」を提案してきた。このグループ分け反転授業の方式を 2017 年度後期における 16 週間の実授業「基礎プログラミング実習」に適用し、自習後の理解度と対面授業後の理解度の観点でその有効性を示してきた。この反転授業の実授業への適用の中で、Java 言語用編集履歴可視化システムを活用し、2017 年度、2018 年度、2019 年度の 3 年間で膨大な学習ログを蓄積した。

5.3.3 分析結果

授業中の課題を解くプログラムを作成する過程で「ビルド&実行」ボタンを押す毎にその時のソースプログラムがログとして蓄積される。今回の分析は、蓄積された 203,436 個（2017 年の 72,193 個、2018 年の 61,721 個、2019 年の 69,522 個）のソースファイルを対象に分析を実施した。各課題を解く際に、一つ前の問題のソースコードをコピーしてそこから修正する学生が多い。その場合には、コピーしたソースコードから大幅に修正を加えてから [ビルド&実行] を行うということが想定される。つまり完成版のソースコードと比較してあまりにも多くの箇所数の修正、あるいはあまりにも長い文字数の修正は、論理エラーとしての対象とするのはふさわしくない。よって「修正箇所数」および「最大修正文字数」が 10 より大きいソースコードは論理エラーとしての検出対象外とした。表 1 に論理エラー種別毎の検出数と年度の合計に対する割合 (%) を示す。

表 1 論理エラー種別ごとの検出数

種類	検出数 (割合 %)					
	2017		2018		2019	
空白	4,189	(19.94)	2,735	(15.12)	2,934	(14.85)
コメント	124	(0.59)	34	(0.19)	289	(1.46)
文字列	2,616	(12.45)	2,675	(14.78)	2,410	(12.20)
括弧	1,140	(5.43)	1,164	(6.43)	1,228	(6.21)
for 文	2,771	(13.19)	2,223	(12.29)	2,577	(13.04)
while 文	152	(0.72)	108	(0.60)	152	(0.77)
if 文	1,453	(6.92)	1,122	(6.20)	1,384	(7.00)
else 文	49	(0.23)	31	(0.17)	183	(0.93)
println	89	(0.42)	101	(0.56)	93	(0.47)
セミコロン	869	(4.14)	649	(3.59)	649	(3.28)
配列	371	(1.77)	333	(1.84)	454	(2.30)
変数	3,215	(15.30)	2,476	(13.68)	2,390	(12.10)
数字	1,447	(6.889)	2,006	(11.09)	2,313	(11.71)
代入文	279	(1.33)	252	(1.39)	331	(1.68)
式	2,220	(10.57)	2,155	(11.91)	2,333	(11.81)
その他	26	(0.12)	29	(0.16)	40	(0.20)
合計	21,010	(100)	18,093	(100)	19,760	(100)

表 1 より、「空白」が最も検出数が多いが、「文字列」の書き換えとともにプログラミングの理解度としては重要性は低い。プログラミングの制御構造として「for 文」と「if 文」に関する検出数が多くなっている。「while 文」は利用頻度が低いために検出数も少ないと思われる。「変数」や「数字」の書き換えの検出数が多くなっている。これは似たような問題を順次解いていくという大学の授業という性質によるものも含まれていると思われる。「式」の検出数も多くなっている。式は一般的には、数値、変数、演算子、関数などの組み合わせが含まれる。今回の分析では数値と変数は個別に扱い、それ以外が「式」に含まれているため、ここに含まれる論理エラーは更なる詳細な分析が必要かもしれない。また、「その他」に分類されたものはコンパイラはエラーを出力しなかったがシフト JIS 全角 2 バイト文字による修正が含まれたソースコードが検出された（コンパイラがエラーを出力しなかった理由は不明）。

なお、表 1 の論理エラーの分類は、我々が大量のソースコードの差分情報を目視で確認した経験に基づく分類である。他の方法としては、対象言語の構文規則を元に分類する方法なども考えられる。

5.4 脳波計測システム

我々は、図 2 の破線の部分の脳波情報を計測する「学習者を感じるサーバ」の開発を行った [20][21]。本節では、システムの概要について述べる。

5.4.1 概要

従来の簡易脳波計を用いた学習時の脳波計測方法では、脳波の計測開始や終了に関して実験参加者本人あるいはスタッフによって人手で行う必要があった。また、脳波シグナルが弱い場合も気づかず、データが取れていない場合もあった。さらに、各実験参加者毎にそれぞれ計測を開始する必要があるため、開始と終了が各実験参加者でずれてしまうという欠点があった。このような従来の脳波データ取得方法の欠点を克服するために脳波収集システムを開発した。

5.4.2 提案システム

前節で示した従来の脳波データ取得方法の欠点を克服するための提案システムを図 8 に示す。この提案システム

により、脳波の計測開始や終了は遠隔管理サーバから指示でき、脳波のステータス確認ができるので実験参加者をサポートする個別のスタッフは不要となり、かつ計測の開始時間と終了時間をすべての実験参加者でそろえることができる。また全脳波計のステータスを確認でき、確認後に脳波の取得開始を指示できるので、脳波がうまく取れていない（シグナルが弱い）ことに気づかず、実験データが取れていないというミスがなくなる。

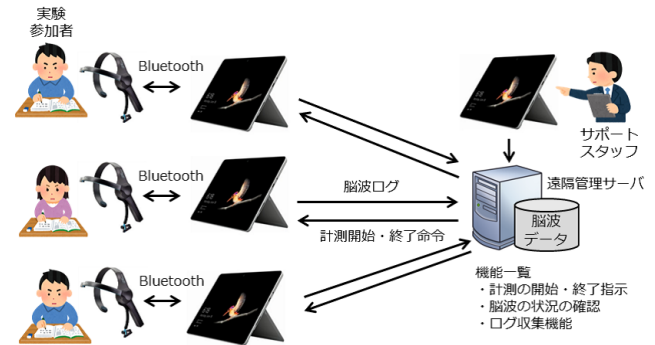


図 8 提案する脳波データ取得システム

6. まとめと今後の課題

本研究では、言語学習を対象とした自学自習システムの研究に関する研究課題「(a) 人工教師を搭載した自学自習システムの開発」、「(b) 英語およびプログラミング言語を対象とした実証実験による評価」、「(c) 英語とプログラミング言語という相異なる言語の学習履歴の統合的分析」、「(d) 普及に向けての非装着型の計測器での代用可能性追求」のうち、(a) に関する研究成果を述べた。今後は「(b) 英語およびプログラミング言語を対象とした実証実験による評価」「(c) 英語とプログラミング言語という相異なる言語の学習履歴の統合的分析」「(d) 普及に向けての非装着型の計測器での代用可能性追求」について研究を進めていく予定である。

謝辞

本研究は JSPS 科研費 JP21K18535, JP20K03082, JP19H01721 の助成を受けたものです。また、本研究の一部は、早稲田理工研特別勘定 1010000175806 NTT 包括協定共同研究、および、経営情報学会「ICT と教育」研究部会の助成を受けたものです。本研究成果の一部は早稲田大学理工総研プロジェクト研究「次世代 e-learning に関する研究」の一環として行われたものです。

参考文献

- [1] 三石大, 今野文子, 長谷川真吾, “復習教材としての授業収録ビデオの LMS による配信と印象評価,” 大学 ICT 推進協議会 2015 年度年次大会論文集, 1D4-4, 2015 年 12 月.
- [2] 重田勝介, “反転授業 ICT による教育改革の進展,” 情報管

- 理 vol.56, no.10, pp.677-683, 2014 年.
- [3] 中学校学習指導要領 (平成 29 年告示), 文部科学省, 2017.
- [4] 高等学校学習指導要領 (平成 30 年告示), 文部科学省, 2018.
- [5] 文部科学省, “新たに取り組むこと, これからも重視することは?”, 平成 29・30 年改訂学習指導要領のくわしい内容, 2017 年 3 月.
https://www.mext.go.jp/a_menu/shotou/new-cs/1383986.htm
- [6] 大森隆行, 丸山勝久, “開発者による編集操作に基づくソースコード変更抽出,” 情報処理学会論文誌, Vol.49, No.7, pp.2349-2359, 2008.
- [7] 森一樹, 田中昂文, 橋浦弘明, 樫山淳雄, 古宮誠一, “プログラミング演習支援のための細粒度履歴収集環境の開発,” 情報処理学会研究報告 (SE), 179(16), pp.1-6, 2013.
- [8] 良峯 徳和, “脳波の謎: リズムとその存在理由,” 多摩大学研究紀要, p.p. 93-100, (2017).
- [9] Hans Berger “On the electroencephalogram in man,” Archiv fur Psychiatrie und Nervenkrankheiten, 87, p.p. 527-570, (1929).
- [10] D. Giannitrapani, “The role of 13-hz activity in mentation,” The EEG of Mental Activities, p.p. 149-152, (1988).
- [11] Kouji Yoshida, Yuuta Sakamoto, Isao Miyaji, Kunihiro and Yamada, “Analysis comparison of brain waves at the learning status by simple electroencephalography,” KES’2012, Proceedings, Knowledge-Based Intelligent Information and Engineering Systems, p.p. 1817-1826, (2012).
- [12] K. Umezawa, M. Nakazawa, M. Kobayashi, Y. Ishii, M. Nakano and S. Hirasawa, “Research and Development Plan of Language-Learning Self-Study System that can Detect Learners’ Conditions over Time and Space,” Proceeding of the 18th Hawaii International Conference on Education, p.p. 402-404, 2020 年 1 月.
- [13] 梅澤克之, 中澤真, 石井雄隆, 小林学, 中野美知子, 平澤茂一, “簡易脳波計を用いたプログラミング学習時のケアレスミスの検出,” 電子情報通信学会 教育工学研究会 (ET) 予稿集, p.p. 135-140, 2020 年 3 月.
- [14] K. Umezawa, M. Nakazawa, M. Kobayashi, Y. Ishii, M. Nakano and S. Hirasawa, “Detection of Careless Mistakes during Programming Learning using a Simple Electroencephalograph,” Proceeding of the 15th International Conference on Computer Science and Education (IEEE ICCSE 2020), p.p. 72-77, 2020 年 8 月.
- [15] 梅澤克之, 中澤真, 後藤正幸, 平澤茂一, “学習履歴を活用したデバッグ練習問題抽出システムの開発,” 第 18 回情報科学技術フォーラム (FIT2019), 予稿集 Vol.3 pp.331-334, 2019 年 9 月.
- [16] K. Umezawa, M. Nakazawa, M. Goto and S. Hirasawa, “Development of Debugging Exercise Extraction System using Learning History,” Proceeding of the 10th The International Conference on Technology for Education (T4E 2019), p.p.244-245, 2019 年 12 月.
- [17] 久保達彦, “diff の動作原理を知る〜どのようにして差分を導き出すのか,” Software Design 2009 年 6 月号, 2009.
- [18] 梅澤克之, 中澤真, 小林学, 石井雄隆, 中野美知子, 平澤茂一, “プログラミング学習時の学習履歴を活用した論理エラーの分析,” 電子情報通信学会 教育工学研究会 (ET) 予稿集, p.p. 41-46, 2020 年 9 月.
- [19] K. Umezawa, M. Nakazawa, M. Kobayashi, Y. Ishii, M. Nakano, and S. Hirasawa, “Analysis of Logic Errors Utilizing a Large Amount of File History During Programming Learning,” Proceeding of the IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE2020), p.p. 232-236, 2020 年 12 月.
- [20] K. Umezawa, M. Nakazawa, M. Kobayashi, Y. Ishii, M. Nakano and S. Hirasawa, “Development of Electroencephalograph Collection System in Language-Learning Self-Study System That Can Detect Learning State of the Learner,” Proceeding of the International Conference on Higher Education Learning, Teaching and Pedagogy (ICHELTP 2020), p. 122, 2020 年 9 月.
- [21] 梅澤克之, 中澤真, 石井雄隆, 小林学, 中野美知子, 平澤茂一, “言語学習を対象とした自学自習システムの研究〜関連研究と脳波収集システムの開発〜,” 電子情報通信学会 教育工学研究会 (ET) 予稿集, pp. 7-12, July 2021.