

Vol. 62

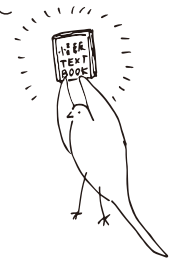
CONTENTS

- 【コラム】情報教育への新しい期待… 川合 慧
【解説】プログラミング教育における反転授業の試み… 高井 久美子 水谷 晃三
【解説】記述式解答の採点支援・自動採点に向けて… 中川 正樹 平井 佑樹

COLUMN



情報教育への新しい期待



次期学習指導要領では情報の扱いが大きくなる。プログラミングへの言及も重みを増している。また、情報学分野の参照基準も作られ、時代が大きく動いていると感じている中で、若干の振り返りを試みたい。

かなり前のプログラミングシンポジウムの席でこんな話を聞いた。「今の大型機のOSは、新宿の街中を歩いている人に「作ってみませんか」と声をかけて集めたような体制で作っている。情報の教育も必要に応じてやればよい」。また、30年前に当時の文部省に計算機教育の重要性を説きに行った場で、担当者に「私が受けたPL-1の2週間講習程度で十分では？」と言われたこともあった。初中等における体系的な教育の話など皆無に等しい時代であった。

高等学校における教科「情報」が開始された年の学会誌に、情報教育が「伝統的基礎学問ではない」という理由で疎外されている状況を嘆いた記事を書いた。教科「情報」については、「伝統的基礎学問」を担当する先生方と高等学校関係者の両方から、ありとあらゆる疑問や反対意見が寄せられていた。そのような中で、情報分野の著名な先達のお一人からこの記事に対するご意見をいただいた。いわく、量（問題規模）の問題は物理学には適わない、高等学校のどの科目を情報の代わりに追放するのか、モデル化には経験が必要なので情報の教育は大学に入ってからがよい、情報の偉い先生方は社会への働きかけもせずに過ごしている、等々であった。

筆者は、まだ情報関連学科もまばらな時代を過ごした。尊敬すべき先達は数多く、その方々なくしては今日の（情動的）状況になり得なかったことは明らかである。このような一騎当千の先達の方々はほとんど独力で道を開けてきたが、その過程はまさに自助努力そのものであったが故に、後進に対する体系的な教育そのものを考える暇もほとんどなしに活動されたであろうことは想像に難くない。個人的感触ではあるが、本会の中でも、このような事情を背景とするとと思われる情報教育軽視の風潮が強かったように思う。

本会でも、1988年に「コンピュータと教育研究会」が、1998年に「情報処理教育委員会」が、それぞれ設置されて以降、教育の問題にも本格的に取り組む体制が整備され、情報教育に真摯に向き合い活動を続けている若い研究者も多くなってきた。近年の、特に情報に関連する委員会と研究会に集っている多くの若い方々の活動に敬服し、大きな期待を持つものである。

川合 慧(放送大学)

プログラミング教育における反転授業の試み

高井久美子 水谷晃三

帝京大学

反転授業への期待

「教える」から「学ぶ」へと教授学習パラダイムの転換の流れを受け、教育の質的転換に向けて、高等教育機関ではアクティブ・ラーニングに深く関心が寄せられている。アクティブ・ラーニングや反転授業といったキーワードがニュースで取り上げられており、世間一般でも広く注目されていると感じている。反転授業とは、授業時間内と授業時間外で行う学習活動を入れ替えた授業形態である。一般的な授業では授業を受けてから宿題に取り組む。反転授業では、授業の前に学習内容に関する知識を学び、授業中にはそれを応用して問題を解くといった活動を行う。たとえば、授業の前に講義ビデオを見ておき、授業中に課題に取り組むといった具合で、これまで宿題としてやっていた「問題を解く」ことを授業時間中にやり、授業時間中にやっていた「講義を聞く」ことを宿題としてやるのである。

反転授業は、これまでは宿題であった課題に1人で取り組む代わりに、授業時間内に教室で先生や仲間と一緒に取り組むことができるところに特徴がある。「本当に教員の存在が目の前に必要になるのは、勉強につまずいて個別の手助けをもとめているときだ。大勢に向けて内容を喋るだけなら、教室で対面する必要はない」というアイデアから始まったことであったと Aaron Sams はその著書「反転授業」¹⁾で述べている。

本稿では、プログラミング教育において反転授業を行った事例を報告する。プログラミングの学習は知識を学ぶだけでなく、実際に知識を活かしてプロ

グラムを作ることが重要であるが、プログラミングの学習を始めたばかりのころは実際にプログラムを作ろうとしたときに小さな問題が障壁になり、先に進めなくなってしまう受講者が見受けられる。個別の手助けが必要なときにこそ教室内で対面して対応ができればと考え、筆者らはプログラミング科目で反転授業を実施することにした。すなわち、受講者は事前学習として講義ビデオを視聴して学習すべき内容を学んでから授業に臨み、授業中はプログラムを作る課題に取り組むといった形態である。本稿では授業設計や授業準備を中心に授業実践について紹介する。授業実践の詳細については、文献2)をご参照いただきたい。

プログラミング教育への反転授業の導入事例

□ 導入の背景

筆者らの学科では大学1年生の前期必修科目の1つとして初学者向けのプログラミング授業を設定している。表-1は本授業の授業計画であり、Javaを用いた演習を含む内容となっている。プログラミングの基本的な概念、接続、条件分岐、繰り返し処理の記述と実際のプログラムの振舞いをそれぞれ理解させることを主な学習目標としている。

プログラミングにおいては、プログラムのソースコードを作成する作業やプログラム実行時の実際の動きなど、教科書などテキスト中心の教材では十分に説明できない内容もある。これを補うため、これまでは授業時間中において15～20分程度の講義を行い、その後に課題プログラムの作成演習を行う授

業構成としていた。この講義時間をなくして演習時間を多く取るための方策が講義ビデオを用いた反転授業の導入である。

□ 反転授業の準備

● 反転授業の導入の方針と講義ビデオの構成

本事例の反転授業では事前学習用の教材として講義ビデオを用意した。反転授業を初めて経験する受講者がほとんどであることと、講義ビデオの事前視聴の負担に配慮して、15回の授業のうち学習内容が新しい内容に切り替わる授業回(表-1中☆印)を反転授業とした。第5回の授業(※印)はプレ反転授

授業回	内容
1, 2	プログラミングの概要, Java の特徴, 基本的な文法
3	プログラム作成環境, 例題プログラムの作成と実行
4, 5 ※	図形描画プログラムの作成
6 ☆	変数と演算
7 ☆, 8	繰り返し処理 (While と for), 多重ループ
9	中間試験と振り返り
10 ☆, 11	条件分岐処理, 論理演算子, 繰り返し処理との組合せ
12 ☆	文字列の取り扱いと表示
13 ☆, 14	総合演習 (Calendar クラス, 型の変換, Math クラス, 複雑な図形)
15	期末試験とまとめ

※反転授業の説明と動画配信システムの操作に慣れてもらうため、第5回授業もプレ反転授業として実施した。

表-1 反転授業を導入した授業計画

業と位置付けて、反転授業の受け方や講義ビデオの視聴方法などを説明する機会を設けた。

講義ビデオの内容は、従来授業時間中に行ってきた講義をPCの画面キャプチャソフトウェア(スクリーンキャストツールなどともよばれる)を用いてビデオ化したものであり、1本あたり10~15分程度の長さである。図-1に示すように、画面上でLMSとよばれる学習支援システムで提供している教材や実際のソースコードなどを表示し、マウスポインタで示しながら音声で解説する構成とした。

● 講義ビデオの作成

講義ビデオの作成にあたり、以下の機材を用意した。

- 収録用のPC(授業で使用するものと同等の開発環境をインストールしたもの)
- 画面キャプチャソフトウェア
- マイク(音声収録向けの専用のもの)

本事例においてはMicrosoft WindowsがインストールされたPCを用意し、画面キャプチャソフトウェアには同社製のExpression Encoder 4(以下EE4と略す)を使用した。同種のソフトウェアは数多く存在するので使いなれたものを使用するのがよいだろう。

具体的な収録手順は次のとおりである(図-2)。まず、①EE4上で映像の入力ソースを画面とし、音声の入力ソースをマイクに設定する。次に、②出力先の設定としてファイルをアーカイブするよう

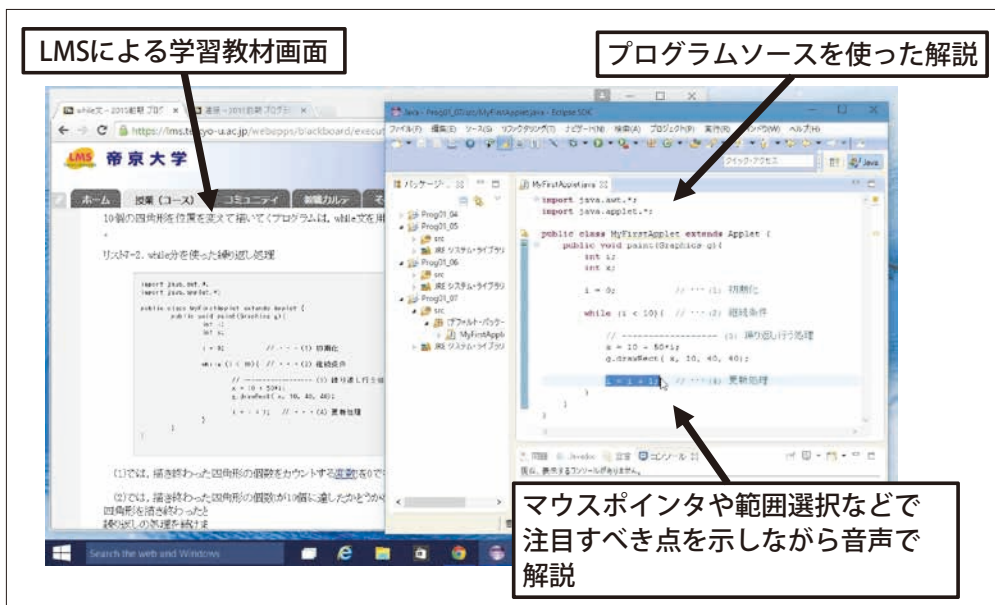


図-1 講義ビデオの画面例



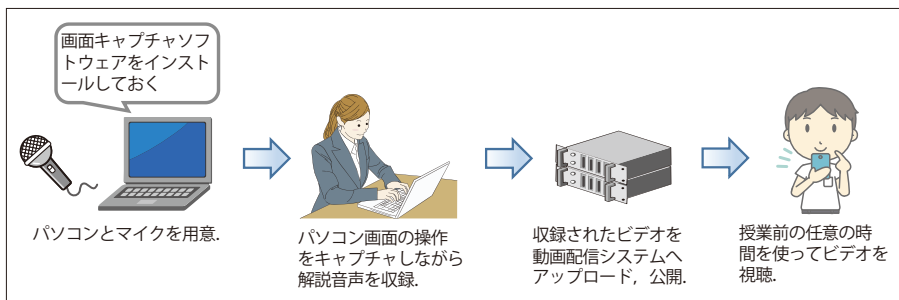


図-2 講義ビデオの準備から受講者への公開までのステップ

に指定する。以上の設定ののち、③開始ボタンを押して収録を開始する。再びEE4上の停止ボタンを押すまで、画面上の操作とマイクの音声と同時に1つのビデオファイルとして記録される。

本事例では、収録後にオーサリングソフトウェアを使用して、字幕を加えたり、不要と思われる内容をカットしたりするなどの編集を行った。完成した動画を動画配信システムにアップロードし、受講者が視聴できるようにした。

● 解説時の台詞について

収録後の編集作業を伴わなければ講義ビデオの作成にかかわる時間は実際の講義に要する時間と大きく変わらない。しかし、収録後の動画を確認してみると、説明の段取りが悪かったり、何度も言葉を噛んだり、同じことを繰り返し喋っていたりして自分でも気になってしまう。気にせずこのまま受講者に公開してもよいのだが、あらかじめ原稿を作成してこれを基に収録する方法を試みたこともある。

この方法では、まずナレーション原稿を作成する。15分ほどの講義ビデオに対する原稿はA4で3～4ページになった。次に、この原稿を音読して音声を収録する。それでも言葉を噛んで録り直したりすると、音声の収録は15分では終わらない。その後、収録した音声を再生しながらその内容に合わせてPCを操作して画面をキャプチャし、オーサリングソフトウェアを使って1本の講義ビデオとして仕上げる。以上の作業のため完成には相応の時間がかかる。作業負荷は小さくないがナレーション原稿が残っているため講義内容の見直しはやりやすくなる。

● ビデオ収録上の配慮

本事例ではPC上の画面をそのまま収録するため以下のような点を配慮した。

- 画面上のメニューやソースコードなどの文字が読みにくくならないように画面解像度やテキストのフォントサイズを調整した。具体的には、PCの画面解像度を1280×720ピクセルに設定して動画解像度もこれに合わせて収録(720p/30fps)した。
- 受講者がどのような環境で講義ビデオを視聴するのかをあらかじめ想定し、スマートフォンなど画面サイズが小さい場合でも文字や図形が小さくならないようにした。具体的には、テキストエディタのフォントサイズを大きめに設定したり、視認しやすい太めのフォントを選択したりした。

また、音声の収録にはピンマイクやいわゆるボイカル用マイクなど音声収録専用のものを用いた。PCに内蔵のものやテレビ会議向けの指向性の広いマイクは周囲の環境音が入りやすいためである。

□ 授業の実施と結果

● 授業の実施と動画配信システムの視聴記録の活用

受講者には次回授業において反転授業を行う旨をあらかじめ伝えておき、授業当日は講義を一切行わずに課題プログラムの作成演習に取り掛かるという構成で反転授業を実施した。

動画配信システムには、受講者が講義ビデオを再生した時刻、受講者のID、再生時間などの視聴記録が蓄積されている。この記録を活用することで受講者の動画視聴率が推定できる。動画視聴率の低い受講者には個別にヒアリングを行い、十分な理解ができていない受講者には授業時間内にビデオを視聴させ、理解できたのちに課題プログラムの作成演習に取り掛かるように指示した。なお、授業全体の視聴率は全体で80.0%であり、視聴率の少ない受講者にはすでに理解している内容のた

め視聴しなかったという者もいた。

● 受講者の声

すべての授業が終了した後に反転授業に関するアンケートを行った。その結果、反転授業を「よかった」とする回答が94.6%となるなど好意的な結果が得られた。自由回答には「予習ができるのでとても良い」「理解するまで時間をかけてできる」「これからもこの反転授業を続けてほしい」などの意見があった。一方、「良くなかった」と答えた受講者の自由回答には「動画を見ても難しすぎる」「聞いていても分からない」「画質が悪い」「文字が小さい」などとあり、講義ビデオの内容や構成に改良の余地があることが分かった。

● 教授者側のメリット

教員側の実感として最大の効果は、演習時間中に受講者からの質問に対応できる時間が増えたことである。特にプログラミング教育においては、同じ要件に対するプログラムの書き方は千差万別であるため、受講者個々の書き方、考え方に応じた指導が欠かせない。そのための時間を多く確保できる点において、反転授業の効果はあったと考えている。

講義ビデオを使った反転授業にしたことで、ビデオを表示しながら「この部分の説明が分からない」といった具体的な質問が増えるようになった。そればかりでなく、「この説明の意味は理解できたけども、課題を解くためにどう適用するのか分からない」などのように、ビデオで得た知識を問題解決に適用する過程で生じる質問もみられるようになった。このような、受講者の能動的な学習活動における疑問に応じやすくなることも講義ビデオを用いた反転授業の1つの利点といえる。

今後の展望

プログラミング教育における反転授業について、準備と実施を中心に述べた。問題点としては、事前学習をしてこない受講者への対応と、事前学習として講義ビデオは視聴するものの内容を理解してきているかどうか分からないといった点が挙げ

られる。また、反転授業が主体的な学習につながっているのだろうかという疑問もある。

これらの問題に対して、現在、所属する学科のプログラミング教育に携わる教員が一丸となっており、カリキュラムや授業方法の検討に取り組んでおり、本事例後の授業においては以下のような改善案を実施している。まず、事前学習をしてこなかった受講者は、そのほかの受講者と着席エリアを別にして、授業時間中に事前学習から始めてもらうようにしている。事前学習における理解を確認するために、事前学習の教材として講義ビデオを見ながら書き込むワークシートを準備し、配付している。さらに主体的な学習となることを期待して、林らがプログラミング教育における反転授業でグループで活動を行った事例³⁾を参考に、授業時間中にグループワークの時間を設けて受講者同士の教え合いの機会を作るほか、プログラミングの読み書きスキルについてのルーブリックを用いて、毎授業時間の自分の到達度を受講者自身が自己評価する方法を取り入れている。また、これらの学習活動を細かくフォローできるように、SA (Student Assistant) 1名が6名から8名程度の受講者を担当する少人数担当制を取り入れている。今後も実践を積み重ね、より良い学習に向けた方策を模索していきたいと考えている。

参考文献

- 1) Bergmann, J. and Sams, A. (著), 山内祐平, 大浦弘樹(序文・監修), 上原裕美子(訳): 反転授業, オデッセイコミュニケーションズ(2014).
- 2) 水谷晃三, 高井久美子: プログラミング初学者を対象にした動画教材による反転授業の実践と評価, 情報処理学会研究報告, Vol.2015-CE-132, No.34, pp.1-8 (2015).
- 3) 林康弘, 深町賢一, 小松川浩: プログラミング教育における反転授業の実践と評価, 教育システム情報学会, 第40回全国大会, A2-2, pp.97-98 (2015).

(2016年5月31日受付)

高井久美子 (正会員) kumiko@ics.teikyo-u.ac.jp

宇都宮大学大学院工学研究科修士。博士(工学)。帝京大学医学図書館司書, 同理工学部情報科学技術職員, 助手, 助教を経て, 情報電子工学科およびラーニングテクノロジー開発室講師。

水谷晃三 (正会員) mizutani@ics.teikyo-u.ac.jp

帝京大学大学院理工学研究科修士。博士(工学)。帝京大学医療情報システム研究センター 助教, 講師を経て, 同理工学部情報電子工学科講師。



記述式解答の採点支援・自動採点に向けて

—手書き認識からの挑戦—

中川正樹 平井佑樹

東京農工大学

記述式問題の必要性と情報化

最近、大学入試センター試験（以下、センター試験）の後継である大学入学希望者学力評価テスト（以下、大学新入試）で記述式問題が導入されることが話題になっている。また、その採点をどうするかが、さまざまな立場から議論されている。

学力試験は、それが授業中の小テスト、全国一斉模擬テスト、入試などのいずれであれ、能力や学習成果を測るために不可欠である。しかし、採点には**多大な労力と時間を要する**。また、人による採点には誤った採点やばらつき（以下、これらを採点ミスと総称）もある。そして、採点結果の返却が遅れるほど、受験者の**復習効果も低下**していく。

こうした課題を解決する手段として、大規模一斉試験ではマークシート方式が採用され、中規模の試験では、CBT/WBT (Computer/Web Based Testing) が普及し始めている。ところが、現状では、受験者がマークあるいはキーボード入力できる問題に限られ、**考える力や深い理解を問うことができる記述式問題はほとんど除外**されている。また、選択式問題が、子供たちの問題解決行動に副作用を及ぼしているという懸念もある¹⁾。

別の課題として、受験者の進路が決まってだいぶ後になって、**入試の採点ミス**が判明する不祥事が後を絶たない。情報公開にこそ情報技術が活用されるべきなのに、それがなされていない。

これら課題の根本的な解決方法は、受験者の考える力や深い理解を問うことができる記述式問題を出

題し、それに対しても採点支援や自動採点を可能にし、受験者の記憶が新しいうちに採点結果をフィードバックすることである。また、受験者が、IDとパスワードを入力すれば、採点結果を確認し、即座に訂正依頼や質問などができるようにすることである。そこに至るまでにはいくつかの課題を解決しなければならないが、今始めれば、データや技術を蓄積することによって、記述式解答の採点支援、あるいは、ある程度の自動採点まで実現できる可能性は十分ある。

なお、ここに述べる内容は、動的な筆記過程が採集できるオンライン入力のタブレットでは、これから5年ぐらいのうちに現実になりそうなことである。一方、紙媒体から筆跡画像を読み込むオフライン入力では、採点支援が先行し、その次に自動採点が現実になる可能性が十分ある。

観測系としての試験

学習者の測定すべき学習成果や能力を変数 x とし、試験 g を課して解答 $g(x)$ を得る。それに採点 f を施して得点 $f(g(x))$ を求める (図-1)。従来は、 g として、記述式、穴埋め、選択式問題が採用され、 f は人が行ってきた。最近では、 g がマークシートや CBT/WBT、 f が機械照合になってきている。この場合、採点は高速で誤りもほとんどない。これが有効な場面も非常に多い。しかし、学習者の考える力や深い理解を問う観測系としてきわめて不十分であることに異論はなからう。

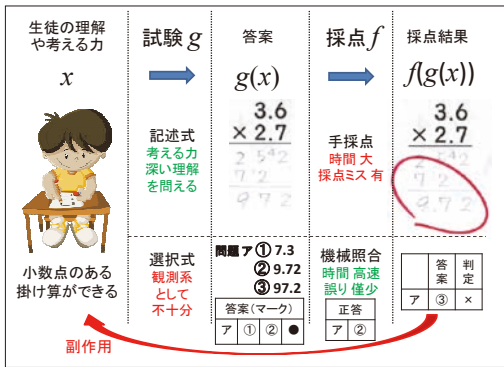


図-1 試験・採点の関数モデル

活用段階	認識確認採点	採点支援	自動採点
概要	学習者がコンピュータの認識結果を確認・訂正	採点者による採点をコンピュータで支援	コンピュータによる採点
媒体	タブレット	タブレット&紙	タブレット&紙
利点	その場で採点と確認、即時フィードバック	採点ミスの防止、質の保証	即時フィードバックと採点労力軽減
要件	正認識率と正採点率の向上、ユーザインタフェース	正認識率と正採点率の向上、誤採点率の低減	誤採点率の低減、誤採点照会や質問への効率的な回答
効果的な利用分野	書写学習、算数・数学採点	ほぼ全分野	採点照会機能と連携すればほぼ全分野*

*) 長文の論述問題と図形の証明問題、グラフや化学式を書く問題などは当面除く。

表-1 情報技術の活用方法の3段階

さらに重要な問題は、観測系が対象に副作用を及ぼすことである。選択式問題を見て、それに解答するに足りる必要最低限だけ問題を読み、ただ選ぶだけの問題解決行動パターンを刷り込んでしまう。これを心身が発達する時期に10年以上もかけてやるのだから、実際の現場で解決策を創案・工夫するのではなく、既存の方法から選択する行動様式になるのも頷ける。事実、選択式問題では、「答えを案ずる」のではなく「答えをあてる」ことができればよい。問題も、選択式にするために回りくどい問い方になっている場合もある。

採点支援・自動採点で何ができるか

記述式解答の採点支援・自動採点がある程度できるとして、どのような利点が得られるだろうか。ここでは、学習者がタブレットなどの筆記と認識ができるデバイスを用いて手書きで問題を解き、その認識結果を確認・訂正して、正しいかどうかの判定を受ける認識確認採点も含めて考察する。

認識確認採点では、手書きの誤認識は採点に影響しない。そのため、キーボード入力難しい数式の解答や時系列情報を判定できる書写学習などで利点がある。後述するように、大規模な試験には向かないものの、小規模の試験や、電車の中などで解答し答え合わせする自習には向いている。このときの認識確認採点への要件は、手書きを正しく認識する正認識率と、正答を正しいと判定し、誤答を間違いと判定する正採点率を上げることである。学習者に

とって認識結果の確認・訂正がしやすいユーザインタフェースの提供も重要である。

一方、採点支援と自動採点では、受験者は受験時に認識結果を確認しない。採点支援の利点は、コンピュータが解答の認識と採点の支援をすることによる採点者による採点ミスの防止と採点の効率化である。採点支援によって、解答をクラスタリング、ランキング、スクリーニングなどして、採点者の採点不均一を防止できる。また、人による採点と機械の採点が違うときに、採点の確認を促すことができる。採点支援への要件は、解答認識の利用方法に応じ、正認識率と正採点率の向上、そして、誤採点率の低減である。複数手法の多数決をとれば、誤採点は減り、採点の信頼性は上がる。なお、機械によるものを誤採点と呼び、採点ミスと区別する。

自動採点は、筆記解答の認識と採点までをコンピュータで行う。その利点は、採点結果を即時に受験者にフィードバックできることと、採点の労力を軽減できることである。そのためには、正答を間違い、あるいは、誤答を正しいと誤判定する誤採点率を極力減らす必要がある。誤採点を避けるために、採点できないと判定する棄却をある程度は許容する。自動採点には、受験者が自分の答案に対する採点を確認し、誤採点があれば照会できる仕組みが前提になり、その照会やその他の質問が多く寄せられる可能性がある。それらをどれだけ低減でき、採点者がどれだけ効率的に回答できるかが重要になる。

これまで述べてきたことを表-1にまとめる。



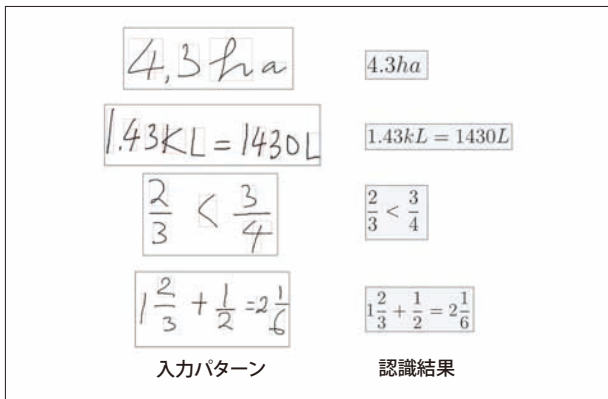


図-2 オンライン手書き数式認識

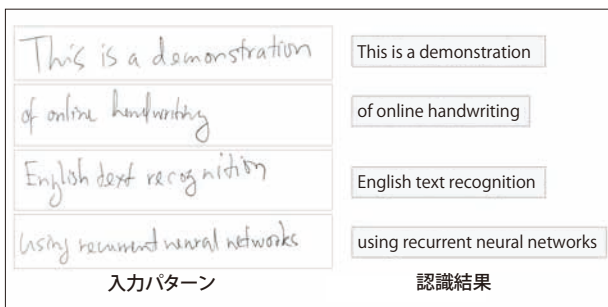


図-3 オンライン手書き英語認識

記述式解答の採点自動化に向けた課題

□ 解答手段

CBT/WBTでは、PC (Personal Computer) の利用が前提になる。選択式問題だけでなく、記述式問題を課すこともできる。受験者はキーボードで答案を入力する。しかし、このときにいくつかの問題がある。まず、数式の入力には向かない。そして、キー入力の音が無視できない。さらに、大規模一斉試験で、すべての受験生に不公平なく同一の環境を用意することは非現実的である。

タブレットでは手書き入力できるため、数式入力にも適する。自習環境では手書き解答を認識させ、結果を確認してから正誤判定できる。図-2, 3には、オンライン手書き数式および英語認識の例を示す。なお、日本語の文章認識は、すでにスマートフォンやタブレットで実用化されている。しかし、PCと同様にタブレットを大規模の一斉試験で利用することは、故障に対する保障や費用などの面から難しい。

以上を踏まえると、紙による記述式試験が最も現

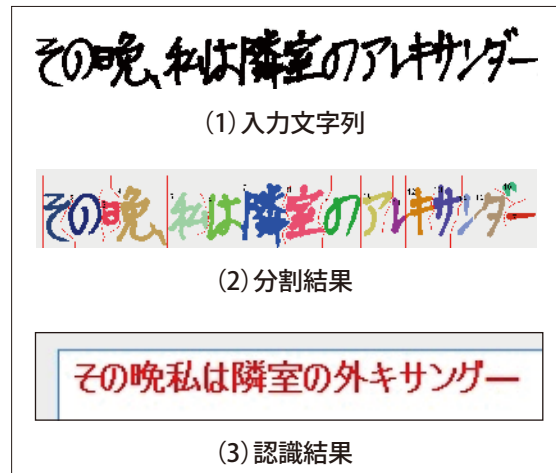


図-4 接触文字のオフライン認識例

実的である。受験者の負担も少ない。

ただし、紙の答案を認識して採点するオフライン認識は、オンライン認識が利用できるタブレット上でのそれより難しい。深層学習が文字認識精度を一段と向上させる可能性があるのは事実であるが、行や文字の切出しや数式などの構造の認識には、まだ難しい課題が残されている。日本語文章で文字が接触する場合に、その分離と認識が難しい例を図-4に示す。そこで、分離すべきところをなるべく取りこぼさない分離方法やそもそも分離しないで文字列を認識する方法の研究が進んでいる。また、大量の学習パターンの蓄積と機械学習の利用によって、手書き認識技術が着実に進歩することが期待できる。さらに、受験者と採点者のユーザインタフェースを工夫することも検討に値する。

□ 緩い採点と厳しい採点

記述式解答の採点では、文字や語句を正しく書いているか否かを厳しく判定する必要がある場合と、多少文字が読めなくても正しいと判定してよい場合がある。たとえば、英単語を書かせる問題では、各文字を正しく書けているかは重要であるが、語句や短文で答えさせる問題では、そこまで要求されることは少ない。採点支援や自動採点ではこうした判断基準の制御が必要になる。

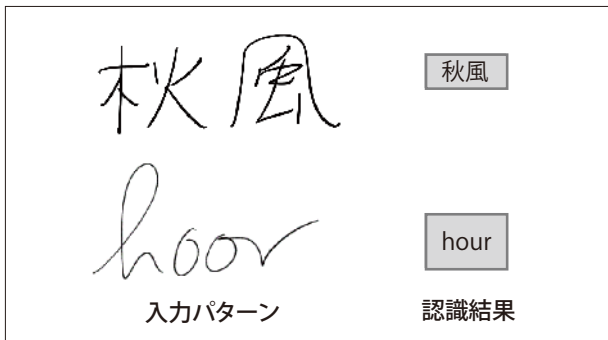


図-5 文脈や辞書による過剰な認識

□ 誤採点の低減

記述式解答の採点支援と自動採点には、解答の認識段階での誤認識と棄却、および、正誤判定段階での誤採点が発生し得る。誤採点が多いと、受験者からの照会が増え、採点者の後作業が多くなる。

現在の手書き文字認識は、文字だけを読むのではなく、言語や語句としての確からしさも考慮するので、多少誤字や脱字があっても認識結果の候補が挙がる。しかし、これが採点では問題になる。単語辞書との照合や文脈処理を用いると、記入した語句が多少間違っても正しい語句に認識されてしまう。また、類似の字形がない文字では、多少の画が抜けても正しい文字に認識されやすい。これらの場合、厳しい採点が阻害される。図-5に例を示す。

誤採点には、正しい解答を間違いと誤判定する偽陰性誤採点と、間違っただけを正しいと誤判定する偽陽性誤採点がある。前者は、照会されることが期待され、採点者の確認を経て訂正されよう。一方、後者は申告されない危険性が高い。したがって、後者は極力低減しなければならない。

□ 同値、同義、含意

採点で注意を要することに同値、同義、含意がある。同値は、たとえば数学では、 $2(x-2)(x-3)$ が正解だとして、 $(2x-4)(x-3)$ 、 $(x-2)(2x-6)$ 、 $2x^2-10x+12$ もすべて正解である。もう少し難しい場合として、 $2\cos^2\theta+\sin^2\theta$ は $\cos^2\theta+1$ と同値である。一般論として、同値判定は多項式時間では解けない。一方、算数で約分して答えよという問題に、たとえば、「 $4/8$ 」と答えられたら、いくら同値といっ

ても、不正解にするか、減点すべきである。

同義や含意は、字面としては違うが、意味として同じか包含関係にある場合である。これらは記述式解答でしばしば発生する。たとえば、「高橋是清は、1936年2月26日に暗殺された」が正解なら、「高橋是清は、1936年2月26日にクーデターを起こした青年将校に射殺された」も正解とすべきだろう。したがって、本質的には自然言語解析が必要である。

□ 小論文

小論文の自動採点は、キーボード入力を対象に、欧米では実用化され^{2), 3)}、日本でも石岡らの Jess⁴⁾ がある。小論文の採点は、内容を問うものではなく、修辞、接続、語彙の適正などを問うことが主である。タブレットでは、文字認識を確認・訂正した上で利用できるであろう。紙上の手書き解答に対しては、手書き文章認識との組合せで、相互補完関係を構築できるか否かが鍵になるだろう。

□ 採点照会機能

採点照会機能は、採点支援・自動採点の採否にかかわらず必要である。人による採点でも、その結果を各受験者に提供すべきである。採点支援の場合は人による採点に近いだろう。自動採点では、迅速に採点を受験者に返すことで受験者の記憶が新しいうちに復習を可能にできる。反面、誤採点の危険に対して、受験者と採点者が解答を共有し、受験者からの問合せに対して、採点者が採点を確認・訂正する、あるいは、質問に答えられるようにすべきである。

採点支援・自動採点に向けての戦略

解答手段は、利用面の容易さを重視し、また、大学新入試を視野に入れ、紙への手書き解答を想定すべきと考える。これから始まる段階で、大量の解答パターンが集積されれば、機械学習に利用できる。また、その認識技術や採点技術は、タブレット入力にも利用できる。

緩い採点と厳しい採点の基準は、科目と求められ



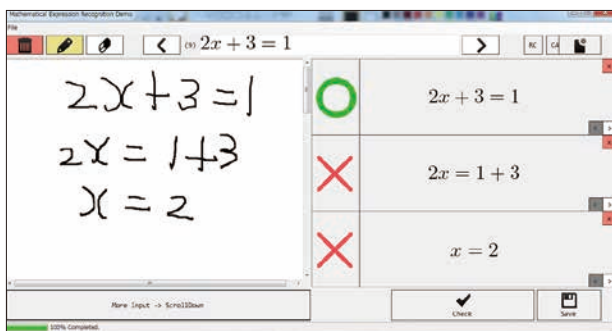


図-6 数式展開の正誤判定

ている解答からある程度自動で設定できそうである。また、採点者が基準を設定できるようにすることもできる。

誤採点低減のためには、基本的に確度の高いものは自動採点を採用し、確度が低いものは採点者に回せばよい。しかし、それが多すぎると自動採点ではなくなる。自動採点のためには、偽陰性誤採点率と偽陽性誤採点を極力ゼロにしなければならない。それぞれが一定以上の認識性能を持つ異質の認識手法を複数採用し、全数一致を求めたり、十分安全な過半数をとったりするなどの戦略が求められる。この中には、「認識しすぎ」を防止するために語彙辞書や文脈を含まないものを加えるのも必要であろう。サンプルパターンを収集して、メーカーや大学に呼び掛けて、コンテストをすることも考えられるし、そこで残った上位によってコンソーシアムを構成し、それらの多数決や全数一致をとれば、信頼性は上がるだろう。

同値、同義、含意への対応として、同値判定のために、算数・数学の自動採点のプロトタイプで数式処理システムを利用したこともある⁵⁾。しかし、そこまでしなくても、何が正解で、何が許される範囲の正解か、減点すべき解答は何で、そのときに何点減点するかなどを指定しておくことで、多くの場合は対応できるように思われる⁶⁾。図-6は、タブレットでのオンライン認識の例であるが、数式の展開のどこで誤ったかを自動検出している。

小論文で、完全な認識と完全な自然言語解釈はできなくても、キーワードの特定やある程度の構文解析はできる。一気に、自動採点を試みなくても、採

点支援として、ある程度のランク付けをし、採点者に託することで、採点の不均一を小さくして、採点の効率を上げることが期待できる。

採点照会機能では、採点者の介在は徐々に減ることが期待できる。また、答案全体の統計を活用したり、受験者ごとの履歴を利用したりできる。これは単独のシステム、あるいは、LMS (Learning Management System)の機能とすることもできる。

今後の展望

本稿では、記述式問題の採点支援や自動採点の可能性を論じた。記述式解答の試験的な採集、自動認識・自動採点コンテスト、実際の入試での解答パターン・技術・ノウハウの蓄積などを進められれば、10年後には効果が見えてこよう。試験とその採点の質、そして、効率の向上に向けて、真の情報化が進展することを願ってやまない。

参考文献

- 1) 河原達也 他：安西祐一郎先生(本会元会長)インタビュー、情報処理、Vol.57, No.3, pp.270-277 (Mar. 2016).
- 2) Burstein, J., et al. : Automated Scoring Using a Hybrid Feature Identification Technique, Proc. 17th Int'l Conf. on Comp. Linguistics, Montreal, Canada, Vol.1, pp.206-210 (1998).
- 3) Wild, F., et al. : Parameters Driving Effectiveness of Automated Essay Scoring with LSA, Proc. 9th Int'l Computer Assisted Assessment Conf., Loughborough, UK, pp.485-494 (2005).
- 4) Ishioka, T. and Kaneda, M. : Automatic Japanese Essay Scoring System : Jess, Proc. 15th Int'l Workshop on Database and Expert Systems Applications, Zaragoza, Spain, pp.4-8 (2003).
- 5) 森重湧太, 中川正樹：手書き数式認識を用いた計算過程の正誤フィードバック、情報処理学会研究報告, Vol.2014-CE-126, No.3, pp.1-6 (2014).
- 6) 小西 渉, 佐々木進亮, 松下朋永, レ・ドゥック・アイン, 中川正樹：手書き数式認識を用いた算数・数学自動採点システム、情報処理学会研究報告, Vol.2016-CE-133, No.7, pp.1-7 (2016).

(2016年5月24日受付)

中川正樹 (正会員) nakagawa@cc.tuat.ac.jp

1977年東京大学理学部物理卒業。1979年同大学院修士課程修了。同在学中、英国Essex大学留学(M.Sc. with distinction)。1979年東京農工大学工学部助手。現在、教授。本会、電子情報通信学会、国際パターン認識連盟フェロー。2016年度、文部科学大臣表彰受賞。理学博士。

平井佑樹 (正会員) yhirai@cc.tuat.ac.jp

2007年東京学芸大学卒業。2009年同大学院修了。2012年筑波大学大学院博士後期課程修了。博士(情報学)。同年東京農工大学助教。2016年8月信州大学アドミッションセンター講師(着任予定)。作問学習、協調学習、教育学習支援システムの研究に従事。