ユーザに作問を任せるテストの自動評点手法とその評価

谷口　敦[1]　井上　創造[2]

概要：本研究では，すべてのユーザーが問題を作成できるシステム作成を目指す．現在，多くの大学で e-Learning のような教育システムが採用されているが，その多くの場面で教師のみ，または限られた人間が問題を作成するため，作問の負担が大きい．また，学習者同士がインターネットを介して, 事前に解答が知られている傾向があり，教師が新しい問題を作成し続ける必要性に直面している．そこで我々は，どんなユーザでも作問できれば教師の負担が減らせることができると考えた．しかしながら誰もが自由に作問できる状況では，問題の難しさがバラバラであったり，試験範囲に見合ってない問題が作問される可能性がある．そこで，作られた問題の難しさと，妥当性を評価に反映させるアルゴリズムを定式化した．またシミュレーションを行い再帰的なアルゴリズムが収束することを確認した．そこで，作問ができ問題を解答できる Web システムを作成した．そのシステムを使い, 実際の人間が作問した問題で，提案手法のアルゴリズムにより，正しい評価ができるかどうか, 大学の授業内で実験を行った．その結果，実際にユーザが作問した問題でも難しさと妥当性を考慮した評点結果が算出された．

A Method for Automatic Assessment of User-generated Tests and Its Evaluation

ATSUSHI TANIGUCHI[1]　SOZO INOUE[2]

1. はじめに

近年，多くの教育現場で e-Learning システムを扱っている．その e-Learning の多くが教師のみ，または限られた人間のみだけが問題を作問できる．しかしそれでは問題数の欠如が起こり，教師に過剰な労働が求められ，問題の質も低下する．そこで我々はこの問題を解決するためにすべてのユーザが問題を作問できまた，編集することが出来るシステムを提供すべきである．しかしながらすべてのユーザが自由に作問ができる状況下では，試験内容に見合っていない問題や，難しさにばらつきがある問題が作問される可能性がある．そこで我々は，学習者自身が問題を作成する方法を構想し，その際においても適切に問題の難易度と妥当性．また評点を与える手法を提案した [1] ．難しさとは一般的に解答の得点によって決まり，妥当性はユーザの問題に対する理解が目的に沿っているかどうかによる．もし我々が難しさと妥当性から推定する方法を開発できれば，例えユーザが作る問題に非常に難しい問題や，無効な問題が存在したとしても，それらを重要度の低い問題として評価する事ができる．提案された手法は，各問題の評点と妥当性を再帰的に定義し，収束するまで反復的に計算する方法である．文献 [1] ではアルゴリズムが収束することを確認するためにシミュレーションを行い，その結果は難しさと妥当性を評価に反映させるものであった．そこで提案手法で述べたアルゴリズムを，実際に人間が作問した問題で使用できるかどうかの実証実験を行った．その結果，実際にユーザが作問した問題でも難しさと妥当性を考慮した評点結果が算出された．

本稿の構成は以下のとおりである．2 章では背景，3 章で作成したシステム，4 章では提案手法，5 章で実験とその結果，6 章で関連研究，最後に 7 章で本稿をまとめる．

2. 背景

近年，クラウドソーシングの研究数が増加しており，自然言語処理やコンピュータビジョン，ヒューマン・コンピュータ・インタラクションの分野での成功例が増えてい

[1] 九州工業大学 大学院 工学府
[2] 九州工業大学 大学院 工学研究院

る．また現在，教育の種類には様々なものがあり，グローバル教育のような少数の民族または発展途上国の人の支援をするものや多くの大学や高等教育機関で導入されているe-Learning システムなど多くの種類がある．教育は教師，教室，カリキュラム，評価などの要素によって成り立っているが，教室を除くこれらの要素が e-Learning の中に存在している．e-Learning は教師が出した問題を生徒が解く空間があるがそれだけではなく生徒同士でコミュニケーションを取れる空間もある．

経験を共有させる方法として Yahoo Answers[17] のようなユーザが他のユーザに質問し，それに対してオープンに議論，コメントする事ができるシステムがある．これはいいアイディアだが自由度が大きく，完全にユーザの経験に依存する．また他の経験を共有させる方法としては自由に動画をアップロードできる YouTube[*1] がある．これもまた経験を共有させるいいアイディアである．しかしながら問題点もある．e-Learning の多くが教師のみ，または限られた人間しが問題を作問できない．教師のみ，または限られた人間のみが問題を作問する場合，教師は過剰に労働しなくてはならない．また質問数の数が足りずランダム性が不足することによる問題の質の低下にもつながる．さらに今日のインターネットの時代では，ユーザ同士がインターネットを介して問題が事前に知られている傾向があり，教師が継続して新しい問題を作成する必要性も直面している．

3. 提案手法

今回の我々の研究目標は, 通常の教育である e-Learning と, ユーザ間で経験を共有することを反映させるようなシステムを作成することである. 提案するものにおいて通常の教育システムと異なる点は

- 教師や限られた人間だけではなく，ユーザ全員が問題の作問者になれる点
- また問題作問可能なユーザは，学生からフィルタリングし登録されるので，教師などの学校の人間は自分の仕事を減らすことが出来る点
- 責任は通常，問題を作問する教師にあるが，今回提案するものでは問題とその答えにある点

である．そこで今回のシステムの要件を以下にまとめる．

3.1 要件

- すべてのユーザが問題を作ることができる
  テストはユーザによって答えられる問題で構成される．特別な課題としては，どのユーザでも問題を作成できるようにすることである．もし，教師のみもしくは限定された人のみが問題作成の許可を持っている場合，教師の過剰な労働，もしくは問題数が少なくランダム性にかけてしまう．また現在のインターネット時代においてユーザ同士が問題を共有し知られてしまうため，教師は新しい問題を作問し続けなくてはならない．これらの問題を解決するためにすべてのユーザが問題を作問でき，自分が作問した問題は編集出来るようなシステムを提供すべきである．
- 別のユーザが問題に答えることができ，自動的に採点される．
  ユーザはウェブブラウザを通じて問題に答えることが出来る．その答えは問題を作問したユーザが定義した正解に従い自動的に採点される必要がある．紙で答えられた問題の採点は範囲外であるが，答えが用意された選択問題に対しては対応できる．その他の答えの採点方法に関しては，少なくとも客観的に何らかの方法で採点されていることを前提とする．

このような状況で，システムは以下の 3 つを自動的に算出する必要がある．

- 問題の難しさ
  ユーザが作成した問題について難しすぎても簡単すぎても採点する際に適切で無いと困る．難しさが適切なレベルの問題を使用する必要があり，問題の難しさは自動的に既存の答えとそれらの採点に基づき推定される必要がある．
- 問題の妥当性
  問題はテストの範囲外のものが使用されるリスクがある．問題が学習目的に沿っている必要がある．例えば語学用のテスト内での数学の問題がある場合や，お楽しみ要素が入っている問題などで，このような無効な問題はいつも役に立たないことはないが，テストはユーザを理解させるための問題で構成されるべきである．その問題の妥当性が問題作成者の振る舞いなどに基づき自動的に推定されることが求められる．
- 評点
  上記の難しさと妥当性の様々なリスクを考慮すると，ユーザの直接採点をユーザの評価を直接採点することは公平でない．ユーザのそのコンテンツの問題の理解度，また能力が関わってくるはずだからである．そこで評価は問題ごとの難しさ，妥当性を考慮し更新していく必要がある．

3.2 評点計算方法

この節ではユーザが生成するの各概念を定式化し，自動評点システムをモデル化する．最後に評価計算をするため反復アルゴリズムを記述する．

以下では，U をユーザ集合，Q を問題集合，q を問題，u を回答者とする．また，問題 q に回答者 u が回答して自動的に採点されたスコアを $score(q, u)$ と表す．スコアは

[*1] http://www.youtube.com

問題や試験ごとに範囲がそれぞれあるが 0 から 1 の範囲にした．

- 問題の難易度
  難易度の高い問題は多くの回答者のスコアが低く，簡単な問題は点数が高いと考えられるため，問題の難しさはその問題のスコアの平均に反比例するものとする．

$$\mathbf{diff}(q) := -\frac{\sum_{u \in U} score(q,u)}{|U|} \tag{1}$$

  ただし，後に式 (3) で乗算される際に難易度が最低の問題は評点を 0 にするため，この値から最小値 $\min_{q \in Q} \mathbf{diff}(q)$ を引いても良い．
- 問題の妥当性
  妥当性はその問題の作問者の評点の平均とする．問題と試験の目標の依存関係のモデル化ができないため妥当性の定式化が少し難しい．従来の授業から推察すると，教師は良い問題を，アシスタントはやや良い問題を，生徒はあまり本格的でない問題を作成するはずである．つまり、評点の高い作問者ほど妥当性の高い問題を作成するものと仮定する．問題 q の妥当性は q の作問者 ($author(q)$ とする) によって解かれた問題の評点の平均と定義できる．問題 q の妥当性は q の作成者によって解かれた問題の評価の平均であると定義する．

$$\mathbf{valid}(q) := \frac{\sum_{q' \in Q} \mathbf{eval}(q', author(q))}{|Q|} \tag{2}$$

  ただし，作問者 $author(q)$ が解いた問題が少ない場合は，この値がたまたま良い値となることも考えられるため，t 検定などにより母集団の区間推定を行い，その信頼区間の最小値を用いても良い．
  また，後に式 (3) で乗算される際に妥当性が低すぎる問題は，むしろ不正解の方が評点が高いことが望ましい場合には，妥当性が低すぎる問題について負の値とするために，例えば平均値より 2σ 小さい値が 0 となるように値を調整しても良い．
- 問題へのユーザの解答に対する評点
  問題の難しさと妥当性を考慮した上で回答者に与えられる評価点で，問題 q に対するユーザ u の回答に対する評点は

$$\mathbf{eval}(q,u) := score(q,u) \cdot \mathbf{diff}(q) \cdot \mathbf{valid}(q) \tag{3}$$

と定義する．

ただし，この値は例えば平均値 50，標準偏差 20 となるように正規化されても良い．

3.3 アルゴリズム

難易度，妥当性，評点は前節で定義した．しかしながら，それらの定義は再帰的である．したがって私たちは収束するまで計算を繰り返すヒューリスティックな手法を用いる．

(1) 各 $q \in Q$ について，式 (1) により難易度 $\mathbf{diff}(q)$ を計算する．

(2) 各ペア $(q,u) \in Q \times U$ について，評点の初期値を

$$\mathbf{eval}(q,u) \leftarrow score(q,u)$$

とする．

(3) 各 $q \in Q$ について，式 (2) により妥当性 $\mathbf{valid}(q)$ を計算する．ただし，教師が作った問題のように明らかに妥当な問題については，妥当性を最高値 $\max_{q \in Q} \mathbf{valid}(q)$ とする．

(4) 各ペア $(q,u) \in Q \times U$ について，評点を

$$\mathbf{eval}(q,u) \leftarrow score(q,u) \cdot \mathbf{valid}(q)$$

により更新する．

(5) 収束するまで (3)-(4) を繰り返す．

(6) 各ペア $(q,u) \in Q \times U$ について，評点を

$$\mathbf{eval}(q,u) \leftarrow \mathbf{eval}(q,u) \cdot \mathbf{diff}(q)$$

により更新する．

ここで評価は初期値としてスコアを与えており，ステップ (3)-(4) において収束するまで式 (2) および (3) の $\mathbf{diff}(q)$ 以外の因子を繰り返し計算している．難易度 $\mathbf{diff}(q)$ は繰り返し乗算されるのを防ぐために，最後のステップ (6) においてのみ乗算される．

上記のアルゴリズムを文献 [1] で，各問題の妥当性にばらつきがある場合と難易度にばらつきがある時で計算機シミュレーションを行い，アルゴリズムが収束することと，難易度と妥当性を適切に導くことを確認した．

4. 作問・自動採点評価 Web システム

3 章ではアルゴリズムを紹介し，シミュレーションを行った．それを実運用するために，実際にユーザが作問し，問題を解答するための Web システムが必要である．実際に作問し，問題を解答してもらうための作問・自動採点評価 Web システム，“Atquiz”(図 1) を作成した．

4.1 Atquiz のシステムの要件

システムの要件を下記に示す．

- 問題作成，問題を編集する機能
  ユーザは自由に問題を作ることができ，また自分が作問した問題に限り修正できるようにする必要がある．
- 問題解答できる
  ユーザが問題を解答した後，採点をする．
- 記述式問題，選択式問題を作成することができる
  記述式，選択式の問題であれば自動的に採点できる．

図 1　作問・自動採点評価 Web システム Atquiz

- 解答された問題は自動採点される
  評点は自動的に採点するため，正解か不正解かを自動採点される必要がある．
- 管理者がボタンを押すと問題の難しさと妥当性を考慮した評点が計算される
  一度データを抜き出し，計算機で計算することは煩わしいので，ボタンひとつですぐに計算できるようにする必要がある．
- 問題はランダムに表示される
  多くの問題が作問されるので，問題に偏りがないように解答させる必要がある．
- 問題は制限時間内のみ解くことが可能
  問題を一括で採点するために制限時間を設ける必要がある．
- 一度解いた問題は表示されない
  答えを知っている問題を解くことで正解率が変動してしまうため
- 一度回答された問題は編集，削除できない
  一度回答された問題を編集，削除できてしまうと正しくアルゴリズムが動かないため制限する必要がある．
- どのような問題にエラーが出たかを調べるために解答者が問題にコメントを出来る機能を作成
  問題を作問する際にヒューマンエラーが起きる可能性がある．自動的にエラーを排除できない領域を知る必要が有るためコメント機能を作成した．
- 自分の成績を見ることができる機能
  回答者は自分が作問した問題，回答した問題を見ることができる．また算出した評点を表示させる必要がある．

その際 Web フレームワークである Ruby on Rails を使用した．また hobo という Ruby on Rails の拡張フレームワークも使用した．

4.2　Atquiz の動作

作問・自動採点評価 Web システム，Atquiz の動作を下記に記す．

4.2.1　問題作成時の動作

問題作成時の動作を説明する．

図 2　Atquiz の作問画面

図 3　Atquiz の作問画面

図 4　解答終了時の画面

(1) 図 2 の一番上の問題テキストボックスに問題を書き込む．
(2) その問題が記述式か選択式かを選ぶ．
  (a) 選択肢の場合はチェックを入れず，その下に問題の回答肢を記入する．
  (b) もし記述式問題を作成したい場合は，“記述式ならチェック”にチェックを入れ，答えを書き込む．

例えば答えが「C 言語」のときは 「c」「C」「c 言語」「C 言語」などと半角や全角，その他回答されうる回答を作成する．

(3) 図 3 の右横の+ボタン，-ボタンを押すことで回答肢を増減ができる．

(4) 選択式問題で正しい回答には“正解にはチェック”にチェックをする．なお記述式の場合は”正解にはチェック”にチェックを入れなくても自動的にすべて正解になるように設定した．

(5) それぞれの回答肢には解説を書き込むこともできる．解説がある際は書き込む，その回答肢が選択され回答された際に表示されるようにした．また問題自体の解説を書き込むこともでき，問題回答時に表示されるようにしている．

(6) 問題作成ボタンを押す

4.2.2 問題回答時の動作

以下に問題解答時の動作を説明する．

(1) 問題を解くボタンを押す．

(a) 選択式問題の場合回答肢が表示される．選択肢ボタンを選択する．

(b) また記述式問題の場合は記述入力フォームが表示され，フォームの中に回答を打ち込む．

(2) 回答ボタンを押すと回答がなされる．

(3) 回答がなされた後，正解か不正解かが自動的に判別され，表示される (図 4)．この際解説文がある場合は解説も表示される．

4.2.3 評点時の動作

回答結果は自動的にデータベースに保存される．結果保存テーブルには，回答された日時，試験 ID，問題 ID，回答された ID または回答された答え，正解か不正解かどうか，回答したユーザ ID，問題を作問したユーザ ID が保存される．1 つの回答ごとに 1 つのデータが保存される．回答するだけでは正解か不正解かの情報しか保存されない．そこで結果保存テーブルより統計解析用のプログラミング言語である R を用いて，問題の難しさと妥当性を反映させた評点を計算し，その結果を再びデータベースの方に上書きするようにしている．

5. 実験

今回の評点方法が評価指標であるかどうかを必要がある．難しさ，妥当性がそれぞれ評点に影響しているかどうかを評価する．

5.1 評価項目

以下に評価項目を示す．

- 評点は問題の難しさ，妥当性を考慮したた評価指標かどうか
- 難しさは評点に影響を与えているかどうか
- 妥当性はは評点に影響を与えているかどうか

5.2 授業

提案手法で述べたアルゴリズムを実際の現場で使用できるかどうか，大学の授業内で実証実験を行った．今回実験は“情報処理基礎”という，学部 2 年生向けのプログラミングの基礎を学ぶことができる授業で行った．“アルゴリズムとデータ構造”はユーザが 4 人と少なくデータ数が少ないので今回は扱わない．“情報処理基礎”の授業は 2 つの学科で行われており，各授業一つの試験科目として同じ試験科目に問題を作問してもらい，同じ試験科目から問題を解いてもらうことにしている．

実験は 2015 年 4 月から行いデータは 2015 年前期授業の 4 回，2 クラス分を使用する．問題の内容は前回授業の復習，及び次回授業の予習にした．また問題を解くことのできる時間は授業開始から約 20 分間とし，ユーザはその時間内であれば何問解いてもいいようにしている．

今回使用する授業で得られた作問数，回答数を表 1 にまとめた．

表 1 使用した“情報基礎処理”のデータ数

試験数	4
問題数	1007
回答された数	14379

5.3 結果

実際にユーザが作問した問題を使用し実験を行った結果，難しさと妥当性に影響された評点が算出された．算出する際，作問者が解いた問題が少ない場合があった．妥当性を算出する際に，母集団の区間推定を行い，50%信頼区間を用いた．また妥当性が低すぎる問題で，不正解の方が評点が高い事が望ましい場合がある．その際，妥当性が低すぎる問題を負の値とするために，平均値より 2σ 小さい値が 0 となるように平均値を調整した．また評点を算出する際に平均値 50，標準偏差 20 となるように正規化した．さらに教師が作問した問題の妥当性を最高値である 100 とし，評点を算出した．

グラフ (図 5) は解かれた問題の評点と妥当性を示している．

図 5 の x 軸は解かれた問題の評点，y 軸は解かれた問題の妥当性である．正解した問題は妥当性が上がるに従い，評点が上がっていることがわかる．また正解していない問題は妥当性が高くなれば評点も下がっていることが見受けられる．

図 6 は解かれた問題の評点と難しさを示している．

図 6 の x 軸は解かれた問題の評点，y 軸は解かれた問題の難しさである．正解した問題の中で，難しさが高ければ高いほど，評点が上がっていることがわかる．また，難し

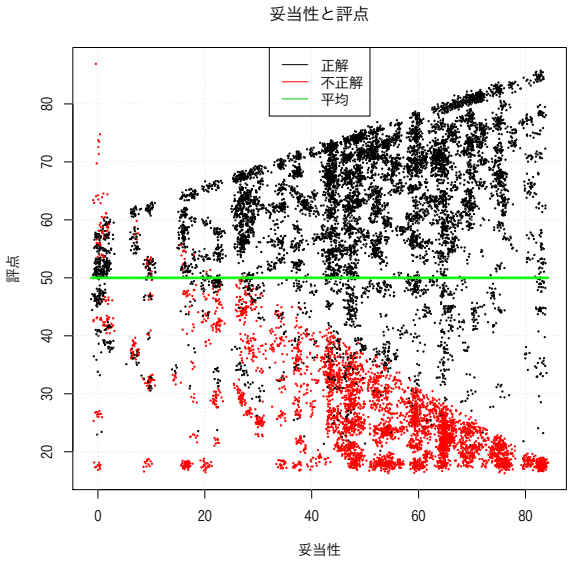

図 5　評点と妥当性のグラフ

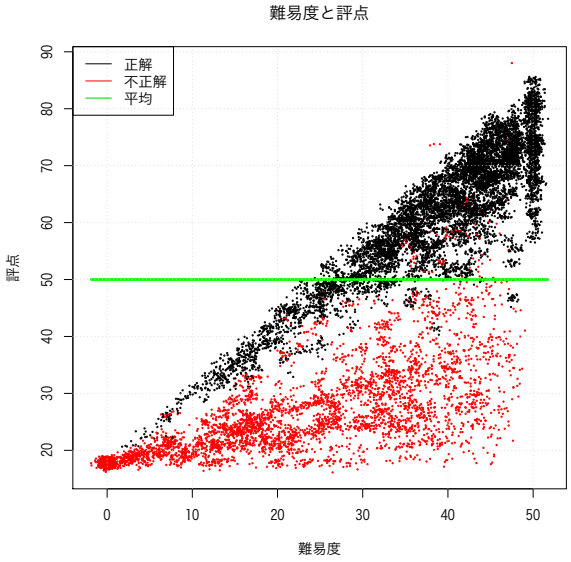

図 6　評点と難しさのグラフ

い問題を間違えた時は簡単な問題を間違えた時よりも評点が高い事が言える．

5.4　考察

グラフに関しての考察を行う．まず，評点と妥当性のグラフ (図 5) において，正解している問題で，妥当性が高く評点も高い問題は「processing 上で色を指定する際，RGB(Red,Green,Blue) 成分の値をそれぞれ 0 から 255 までの値で指定する方法がある一方で，HSV 体系で指定する方法もあります．色相 (Hue)，彩度 (Saturation)，もう一つの V が表わすものは何でしょう？」「Processing において，アニメーションの draw() の中では，キーボードやマウスが入力されたことを何というでしょう？次の選択肢の中から選んでください．1．”コントローラー”，2．”コンソール”，3．”タッチ”，4．”イベント”，5．”コード”」と言うような難しい問題が多く好ましい．

正解している問題で，妥当性が高く，評点が低い問題は「乱数とはなんでしょう．1．心が乱れた時の感情の程度を 1 から 10 までで表した数値　2，常に一つの数値にとどまらずに自由にかわりつづける、一匹数字　3，1 から 108 までの数値の中から任意に取り出した数値　4，ある範囲の数値から任意に取り出した数値」といった問題や「fill(R 成分,G 成分,B 成分) の R,G,B はそれぞれ何の略でしょう．」のような，難しさが低い，簡単な問題が多いことがみられた．問題の妥当性は高いが，難しさが低いため，評点が低いのではないかと考えられる．

妥当性が低く，評点が低い問題は「rect(5,5,3,2); によって描かれる長方形の高さは？」や「集合 a,b,c,d の部分集合の個数を答えろ．」など記述式の難しさも低い問題がみられた．

次に評点と難しさのグラフ (図 6) を見る．

評点も難しさも高い問題には

「`for(i=0;i<100;i=i+5){`
`if(45<i && i<63){println(i);}}`
上記のプログラムが void draw() の中にあるとき，どのように表示されるか，始めの 5 行分を記述してください」という問題や，

「`\int i=0; while(i<4){ println (i); i++;}`の答えは何？」
というような，プログラムから正解を答えさせる問題が多くあった．妥当性が高い問題が望ましい．

評点が低く難易度も低い問題には「ペンの太さを変えるには？」や「プログラムの中で改行を入れたくなった場合、何を書き入れたらいいか書きなさい。」といった記述形式の問題が見受けられた．

正解している問題で，評点が低く，難しさが高い問題は「アルゴリズムの性能を測る概念を何といいますか，1．”熱量”，2．”質量”，3．”計算量”」や，「色を塗りつぶすプ

ログラム fill(); の（）内の要素の順番を次から選べ．1．”赤、青、緑”，2．”赤、緑、青”，3．”緑、青、赤”，4．”緑、赤、青”」と言ったような質問を並べ替えただけのものなど，妥当性が低いと思われる問題があった．難しさが高く，評点が低いため，妥当性が低い問題であることが望ましい．

6. 関連研究

我々の論文は教育分野におけるクラウドソーシングとヒューマンコンピューテーションのためのアプリケーションである．クラウドソーシングとヒューマンコンピューティングはその直接的な関与と人間中心の研究と伝統的なコンピュータサイエンスの両方の信頼が特徴的である．クラウドソーシングとはアウトソーシング *outsourcing* を改変した造語で、インターネット越しに待機している不特定多数の群集 *crowd* にタスクをアウトソーシングするというアイデアを指す言葉である [2]．クラウドソーシングは物理的な世界とネット世界両方に活用されている [3]．ヒューマンコンピュテーションとは“計算機が処理できないタスクの処理に人手を利用する事である．”と，Von Ahn によって 2005 年に書かれた“Human Computation”[4] の中で定義されている．またその他の論文 [4], [5], [6], [7], [8], [9], [10] においても類似の定義が登場する．Google で使用されているような画像検索，成人コンテンツのフィルタリング [11] や常識的な推定 [12], [13]，コンピュータビジョン [14]，またセキュリティ [15], [16] のような分野においても適用されている．本研究は不特定多数の群衆 *cloud* にタスクを任せるという意味で，クラウドソーシングの概念を使い, 教育分野に特化させた．また通常のクラウドソーシングとは異なり, 報酬は与えていない．

7. まとめ

本論文ではどんなユーザでも問題が作問できる Web システムの作成を目指し，ユーザに成績を与える際に問題の難しさ，妥当性を考慮して採点し，それを評点とするアルゴリズムを示した．その際作問・自動評点 Web システム“Atquiz”を作成し実際に大学の授業内でユーザが作問，問題を解答した実験を行った．その結果，実際に作問された問題で，難しさ，妥当性が評点に影響している問題が見受けられた．従来の授業との比較をする必要があるが，教師が作問した問題が少なく，比較ができない状況であるそこで中間テスト，期末テストを教師の問題で行い，提案した採点方法との比較を行う必要がある．また，似た問題や Web サイトからのコピーアンドペーストを発見する機能を作成し，より精度を高めていくことも目標である．

謝辞

本研究の一部は，挑戦的萌芽研究「プロジェクト型学習における自動相互評価方式（研究代表者：井上創造)」および基盤研究 (B)「物理層と意味層の２階層からなるセンサコンテキスト推定技術（研究代表者：井上創造)」による．

参考文献

[1] Ghada Farouk Naiem, 井上創造,“A Method for Assessing User-generated Tests for Online Courses Exploiting Crowdsourcing Concept”, IWWISS, September, 2014
[2] Howe, Jeff. Crowdsourcing: A Definition http://crowdsourcing.typepad.com
[3] Anhai Doan, Raghu Ramakr ishnan, and Al on Y. Halevy. Crowdsourcing Systems on the World-Wide Web. Communications of the ACM 54,4 (April 2011).
[4] von Ahn, L. Human Computation. Doctoral Thesis. UMI Order Number: AAI3205378, CMU, (2005).
[5] Chan, K. T., King, I., and Yuen, M. Mathematical modeling of social games. Proc CSE 2009, 1205-1210.
[6] Law, E. and von Ahn, L. Input-agreement: a new mechanism for collecting data using human computation games. Proc. CHI 2009, 1197-1206.
[7] Law, E., West, K., Mandel, M., Bay, M., Downie, J. S. Evaluation of algorithms using games: the case of music tagging. Proc. ISMIR 2009, 387-392.
[8] Quinn, A. and Bederson, B.B. A Taxonomy of Distributed Human Computation. Tech. Rep. HCIL-2009-23, University of Maryland, (2009).
[9] Schall, D., Truong, H., and Dustdar, S. The Human-Provided Services Framework. CECANDEEE 2008.
[10] Yang, Y., Zhu, B.B., Guo, R., Yang, L., Li, S., and Yu, N. A comprehensive human computation framework: with application to image labeling. Proc. MM 2008.
[11] N2H2, Inc. N2H2 Filter. http://www.n2h2.com
[12] von Ahn, L., Kedia, M, and Blum, M. Verbosity: a game for collecting common-sense facts. Proc CHI 2006.
[13] von Ahn, L. and Dabbish, L. Labeling images with a computer game. Proc CHI 2004.
[14] Barnard, K., Duygulu, P., and Forsyth, D. A. Clustering Art. IEEE conference on Computer Vision and Pattern Recognition, 2001, pages 434-441.
[15] von Ahn, L., Maurer, B., McMillen, C., Abraham, D., and Blum, M. ReCAPTCHA: human-based character recognition via web security measures. Science, 321:5895, (Sept. 12, 2008), 1465-1468.
[16] Pinkas, B. and Sander, T. Securing Passwords Against Dictionary Attacks. In Proceedings of the ACM Computer and Security Conference (CCS' 02), pages 161-170. ACM Press, November 2002.
[17] ”Home — Yahoo Answers.”Yahoo! Answers. Yahoo!, n.d. Web. 26 Mar. 2014.