1E5-04

EMアルゴリズムを用いた教育支援のための自動生成解のエラー除去手法

An approach of removing errors from generated answers for E-learning

中野 智文[*1] 小山 由紀江[*1] 犬塚 信博[*1]
Tomofumi Nakano Yukie Koyama Nohuhiro Inuzuka

[*1]名古屋工業大学
Nagoya Institute of Technology

E-learning has started to attract attention of a lot of people. Although it is convenient for students, teachers need efforts to its create the contents. Therefore we desire a computer system generating questions and answers. However, such a system does not always generate them correctly. As a result, human instructors have to check the generated questions and remove the errors. In this paper, we propose a technique for removing errors in the generated questions. The technique uses EM algorithm in order to estimate the generated accuracy (probability of correct answer) of each question. In the experiment using an E-learning system which generates grammatical questions from English corpus, we can remove errors in the questions.

1. はじめに

近年 E-leanring が注目されている．遠隔授業や，マルチメディアの利用など，さまざまな利点がある．しかしながらそのコンテンツ（教材）の作成が大変なことも指摘されている．それゆえ，問題とその正解をもコンピュータが自動生成するようなシステムの登場が望まれる．ところが，そのシステムの構築には，いつも正解を生成する完全な知識をコンピュータに与える必要がある．計算練習のような限られた分野においては，完全な知識を用意することが可能であるが，一般的には難しいと考えられる．

そのような不完全な知識のシステムにおける問題を解決する方法として，解答の生成を行った後，生成された解答が正解かどうかを人間の教師が確認し，誤りのある問題を除去することが考えられる．しかしこれも結局手間がかかる．本稿では，その誤った解答の問題の除去もコンピュータが行うエラー除去手法を提案する．誤った解答をどのように見つけるかは，Expectation-Maximization(EM) アルゴリズム [2] と学生の解を用いる．EM アルゴリズムは，確率に基づく手法であり，不確定な事象の確率を推定する手法として多く応用されている．事象の期待値の推定とその事象のモデルのパラメータの最尤推定を繰返し行なう．本手法では，学生の解答とシステムによって生成された解答との一致する確率モデルを用いて EM アルゴリズムを適用する．

本手法の有効性を確認するために，英文コーパスからの文法問題の生成システムに対して応用実験を行った．エラー除去前は，システムが生成した正解率が 0.78（サンプルテスト）であったのに対して，誤りと推定された全体の 5 分の 1 の問題を除去した結果，正解率を 0.96 に高めることができた．

次の第 2 節では，生成システムの定義をする．第 3 節では，EM アルゴリズムとその各ステップで必要な期待値の推定方と最尤推定の方法について述べる．第 4 節では，実験とその結果について述べる．

連絡先: 〒 466-8555 名古屋昭和区御器所町 名古屋工業大学 情報メディア教育センター 中野智文, TEL(052)735-5176(直)，nakano@center.nitech.ac.jp

表 1: 生成システムの定義

$\boldsymbol{Q}$	問題の集合.
$\boldsymbol{S}$	学生の集合.
$a_{i,s}$	i 問目おいて s 番の学生解とシステム解の一致. 一致なら $a_{i,s}=1$，不一致なら $a_{i,s}=0$.
$\boldsymbol{A}$	全ての a の集合.

2. 生成システムの定義

正解を生成するシステムの定義をする．問題の集合を $\boldsymbol{Q}$ と表す．システムは問題 $\boldsymbol{Q}$ に対して正解を生成し，この正解をシステム解 とよぶ．学生の集合を $\boldsymbol{S}$ と表す．その学生 $\boldsymbol{S}$ は問題集合 $\boldsymbol{Q}$ に対して解答する．この解答を学生解とよぶ．

システム解や学生解は，それが本当に正解であるかはわからないが，少なくともシステム解と学生解の一致は調べることができる．この両方の解，すなわち，ある i 問目の問題に対してのシステム解と s 番の学生の一致を，$a_{i,s}(i \in \boldsymbol{Q}, s \in \boldsymbol{S})$ として表す．もし $a_{i,s}=1$ ならば一致，$a_{i,s}=0$ ならば不一致とする．この定義を表 1 に示す．

3. EM アルゴリズム

各システム解が正解である確率（正解率）を推定するために，EM アルゴリズムを適用する．各システム解の正解率の集合 $\boldsymbol{\beta}=\{\beta_1,\beta_2,...,\beta_{|\boldsymbol{Q}|}\}$ を推定するために，各学生の正解率の集合 $\boldsymbol{\alpha}=\{\alpha_1,\alpha_2,...,\alpha_{|\boldsymbol{S}|}\}$ も推定する必要がある．システム解と学生の解が一致する確率モデルを与え，それに基づき各学生の正解率を最尤推定する．これを繰り返すのが EM アルゴリズムであり，表 2 に示す．この節では，システム解の正解率の推定と，各学生の正解率を最尤推定について述べる．

3.1 Expectation step

Expectation step では，システム解の正解率 $\beta_i \in \boldsymbol{\beta}$ を見積もる．記号 ^ は一時変数として表 2 で使っている (例えば，$\hat{\beta}_i$ は一時的な β_i の仮の値)．その正解率 β_i は条件付き確率 $p(y_i=1|\boldsymbol{A})$ を使って推定する．ここで y_i は 1 か 0 の値をと

表 2: EM アルゴリズム

1. 入力 $\boldsymbol{A}$.
2. $\hat{\beta}_i = 1.0$ として初期化, その後，$\hat{\alpha}_s$ と $\hat{\pi}$ を Step 4 で得る.
3. Expectation step: for each $i \in \boldsymbol{Q}$

$$\hat{p}(\boldsymbol{A}_i|y_i) = \prod_{a_{i,s} \in \boldsymbol{A}_i} p_{\hat{\alpha}_s}(a_{i,s}|y_i), \quad \text{for each } y_i = 0, 1 \tag{1}$$

$$\hat{\beta}_i = \frac{\hat{\pi} \cdot \hat{p}(\boldsymbol{A}_i|y_i = 1)}{\hat{\pi} \cdot \hat{p}(\boldsymbol{A}_i|y_i = 1) + (1 - \hat{\pi}) \cdot \hat{p}(\boldsymbol{A}_i|y = 0)} \tag{2}$$

4. Maximization step: for each $s \in \boldsymbol{S}$

$$\hat{\alpha}_s = \frac{\sum_{a_{i,s} \in \boldsymbol{A}_s^1} \hat{\beta}_i + \sum_{a_{i,s} \in \boldsymbol{A}_s^0} (1 - \hat{\beta}_i)}{|\boldsymbol{A}_s|} \tag{3}$$

$$\hat{\pi} = \frac{\sum_{i=0}^{|\boldsymbol{Q}|} \hat{\beta}_i}{|\boldsymbol{Q}|} \tag{4}$$

5. 収束するまでステップ 3 と 4 を繰り返す.

り，システム解が正解か不正解に対応する．i 問目の問題以外の学生の解とは無関係（独立）であるので，$p(y_i|\boldsymbol{A}) = p(y_i|\boldsymbol{A}_i)$ である．ここで $\boldsymbol{A}_i = \{a_{i',s} \in \boldsymbol{A} | i' = i\}$ である．この確率は，ベイズの定理を用いて次のようになる．

$$p(y_i|\boldsymbol{A}_i) = \frac{p(y_i) \cdot p(\boldsymbol{A}_i|y_i)}{p(\boldsymbol{A}_i)} \tag{5}$$

ここで $p(y_i)$ はシステム解が正解 ($y_i = 1$) となる事前確率である．また，$p(\boldsymbol{A}_i|y_i)$ は y_i が観測された時の $\boldsymbol{A}_i$ の事後確率である．3つの確率 $p(y_i), p(\boldsymbol{A}_i)$ そして $p(\boldsymbol{A}_i|y_i)$ は，次のように見積もることができる．一つ目，$p(\boldsymbol{A}_i)$ は $\sum_{y_i \in \{0,1\}} p(y_i) \cdot p(\boldsymbol{A}_i|y_i)$ より得ることができ，なぜなら $\boldsymbol{A}_i$ は定数ゆえ $\sum_{y_i \in \{0,1\}} p(y_i|\boldsymbol{A}_i) = 1$ となるからである．2 つ目に，事前確率 $p(y_i = 1)$ が問題によらず一様な値 π になると仮定すると，それは全てのシステム解の正解率の平均として見積もることができる．以上のから式 (5) は式 (2) となる． 3 つ目であるが, その条件付き確率 $p(\boldsymbol{A}_i|y_i)$ は，$\boldsymbol{A}_i$ は独立であるので，次のように求められる．

$$p(\boldsymbol{A}_i|y_i) = \prod_{a_{i,s} \in \boldsymbol{A}_i} p_{\alpha_s}(a_{i,s}|y_i)$$

ここで $p_{\alpha_s}(a_{i,s}|y_i)$ は y_i のときに学生とシステムの解が一致する確率である．この式が，ステップ 3 の式 (1) である．全ての問題上で学生の正解率が一様であると仮定すると，次のようになる．

$$p_{\alpha_s}(a_{i,s}|y_i) = \begin{cases} \alpha_s & : y_i = a_{i,s} \\ 1 - \alpha_s & : y_i \neq a_{i,s}, \end{cases} \tag{6}$$

ここで α_s は s 番の学生の正解率である．

3.2 Maximization step

Maximization step では，学生の正解率 $\boldsymbol{\alpha}$ を最尤推定する．

学生とシステム解の一致する確率モデルを考える．式 (6) より，その一致 $a_{i,s}$ の確率密度は，システム解が正解であるとき $y_i = 0$ と不正解であるときの $y_i = 1$ 2 つの場合の合計である．すなわち，

$$p(a_{i,s}) = \pi_i \cdot p_{\alpha_s}(a_{i,s}|y_i = 1) + (1 - \pi_i) \cdot p_{\alpha_s}(a_{i,s}|y_i = 0)$$

ここで π_i は事前確率 $p(y_i = 1)$ である．$p_{\alpha_s}(a_{i,s}|y_i = 0) = 1 - p_{\alpha_s}(a_{i,s}|y_i = 1)$ と，$p_{\alpha_s}(a_{i,s}|y_i = 1)$ を ϕ_{α_s} とすることにより，この式は，

$$p(a_{i,s}) = \pi_i \cdot \phi_{\alpha_s}(a_{i,s}) + (1 - \pi_i) \cdot (1 - \phi_{\alpha_s}(a_{i,s}))$$

となる．その対数尤度は，

$$\ell(\boldsymbol{\alpha}; \boldsymbol{A}) = \sum_{a_{i,s} \in \boldsymbol{A}} \log[\pi_i \phi_{\alpha_s}(a_{i,s}) + (1 - \pi_i)(1 - \phi_{\alpha_s}(a_{i,s}))]$$

この式を解くのは難しいが，次の対数尤度 ℓ_0 を最大化させることにより ℓ はいつも増加することが知られている．

$$\ell_0(\boldsymbol{\alpha}; \boldsymbol{A}) = \sum_{a_{i,s} \in \boldsymbol{A}} \Delta_i \log \phi_{\alpha_s}(a_{i,s}) + (1 - \Delta_i) \log(1 - \phi_{\alpha_s}(a_{i,s}))$$

ここで $\Delta_i \in \{0, 1\}$ は隠れ変数である．もし $\Delta_i = 1$ ならそのシステム解は正解であり，そうでなければ不正解である．その期待値 Δ_i は，

$$\begin{aligned} \mathrm{E}(\Delta_i|\boldsymbol{A}, \boldsymbol{\alpha}) &= \Pr(\Delta_i = 1|\boldsymbol{A}, \boldsymbol{\alpha}) \\ &= p(y_i = 1|\boldsymbol{A}, \boldsymbol{\alpha}) = p(y_i = 1|\boldsymbol{A}_i, \boldsymbol{\alpha}) = \beta_i \end{aligned}$$

これは，Expectation step の式 (2) によって求められる．期待値 β_i の代入により，

$$\ell_0(\boldsymbol{\alpha}; \boldsymbol{A}) = \sum_{a_{i,s} \in \boldsymbol{A}} \beta_i \log \phi_{\alpha_s}(a_{i,s}) + (1 - \beta_i) \log(1 - \phi_{\alpha_s}(a_{i,s}))$$

この式の最大化は $|\boldsymbol{S}|$ 個の式に分割できる．その中の一つは，

$$\ell_0(\alpha_s; \boldsymbol{A}_s) = \sum_{a_{i,s} \in \boldsymbol{A}_s} \beta_i \log \phi_{\alpha_s}(a_{i,s}) + (1 - \beta_i) \log(1 - \phi_{\alpha_s}(a_{i,s}),$$

となる．ここで $\boldsymbol{A}_s = \{a_{i,s'} \in \boldsymbol{A} | s' = s\}$．さらにその式は，$a_{i,s} = 1$ と $a_{i,s} = 0$ の 2 つの場合に分割できる．式 (6) より，

$$\begin{aligned} \ell_0(\alpha_s; \boldsymbol{A}_s) &= \sum_{a_{i,s} \in \boldsymbol{A}_s^1} \beta_i \log \alpha_s + (1 - \beta_i) \log(1 - \alpha_s) \\ &+ \sum_{a_{i,s} \in \boldsymbol{A}_s^0} \beta_i \log(1 - \alpha_s) + (1 - \beta_i) \log \alpha_s \end{aligned}$$

となる．ここで $\boldsymbol{A}_s^0 = \{a_{i,s} \in \boldsymbol{A}_s | a_{i,s} = 0\}$, $\boldsymbol{A}_s^1 = \{a_{i,s} \in \boldsymbol{A}_s | a_{i,s} = 1\}$．$\log \alpha_s$ と $\log(1 - \alpha_s)$ の項を使って書き直すと，

$$\ell_0(\alpha_s; \boldsymbol{A}_s) = \gamma \log \alpha_s + (|\boldsymbol{A}_s| - \gamma) \log(1 - \alpha_s)$$

ここで $\gamma = \left(\sum_{a_{i,s} \in \boldsymbol{A}_s^1} \beta_i + \sum_{a_{i,s} \in \boldsymbol{A}_s^0} (1 - \beta_i) \right)$ そして $|\boldsymbol{A}_s| - \gamma = \left(\sum_{a_{i,s} \in \boldsymbol{A}_s^1} (1 - \beta_i) + \sum_{a_{i,s} \in \boldsymbol{A}_s^0} \beta_i \right)$．この式を最大化するような α_s の値を求めたい．その微分と 2 階微分の式は，各々，

$$\frac{d\ell_0(\alpha_s; \boldsymbol{A}_s)}{d\alpha_s} = \frac{\gamma}{\alpha_s} - \frac{|\boldsymbol{A}_s| - \gamma}{(1 - \alpha_s)} \tag{7}$$

$$\frac{d^2\ell_0(\alpha_s; \boldsymbol{A}_s)}{d\alpha_s^2} = -\frac{\gamma}{\alpha_s^2} - \frac{|\boldsymbol{A}_s| - \gamma}{(1 - \alpha_s)^2} \tag{8}$$

である．式 (8) は常に負であるので，その対数尤度は式 (7) $= 0$ のとき最大値に達する．それゆえそのような α_s は次式で得られる．

$$\alpha_s = \frac{\gamma}{|\boldsymbol{A}_s|}$$

これが EM アルゴリズム (表 2) 中の式 (3) である．

4. Experiment

次に述べるような我々の E-learning 生成システムに，提案手法を適用した．そのシステムは，大学生用の WBT(Web based teaching) のシステムである．そのシステムは，約 150 万単語からなる将来その大学生たちが読むであろうジャーナルの記事と雑誌論文のコーパスから問題が選ばれ，解答が生成される．問題は文中の主動詞を選択するものである．学生がよく間違えるような文章に的を絞るため，5,868 文がシステムによりあらかじめ精選された．2 つのツールがシステム解を生成するために使われた．一つは Apple PieParser [3] である．もう一つは，Brill Tagger [1] である．これらのツールは完全な処理ができない．表 3 の「合計」列の総合正解率を見てほしい．たったの 0.78 である．218 人の学生がそのシステムに挑戦し，37,333 の学生解を得ることができた．図 1 にそのページの例を示す．

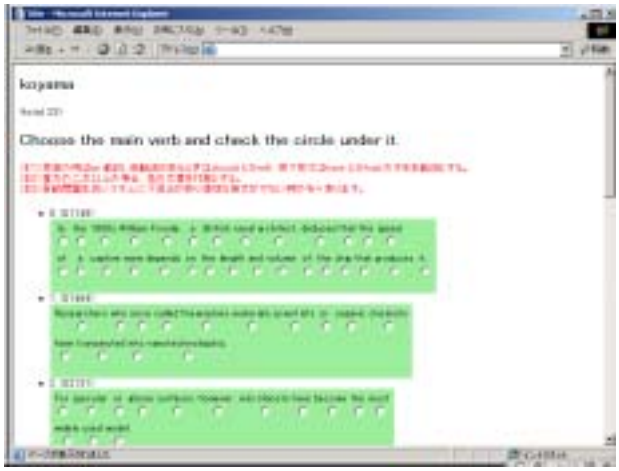

図 1: web で出題されるの問題の例

システム解が誤ったものを除去したい．提案した EM アルゴリズムによってシステム解の正解率を推定した．全ての問題をその推定された正解率によって並べなおした．それを図 2 に示す．推定手法の信頼性を確認するため，サンプリングテスト

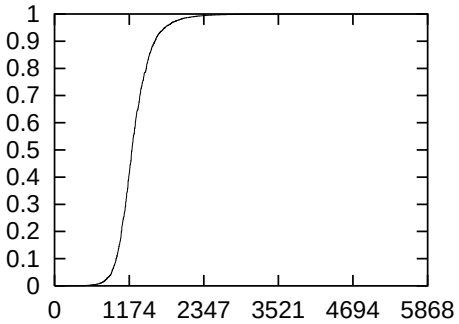

図 2: 推定された正解率．x: 問題の番号 (正解率により並び替えられている)，y: 正解率 β.

を用いた検証を行った．問題の集合を，推定された正解率によりランク付けされた 5 つの集合に，等分割した．それから各ランク集合から 30 問題をランダムに選択し，そのシステム解を英語の教師がチェックした．その結果を表 3 と，図 3 に示す． 図 2 の推定された正解率と図 3 のテストによる正解率両方がよく一致しており，推定された正解率が順序だけでなく，推定値そのものが信頼できるレベルであることが分かる．

表 3: サンプリングテストの結果

ランク番号	1	2	3	4	5	合計
正解数	2	26	30	29	30	117
不正解数	28	4	0	1	0	33
正解率	0.07	0.87	1.00	0.97	1.00	0.78

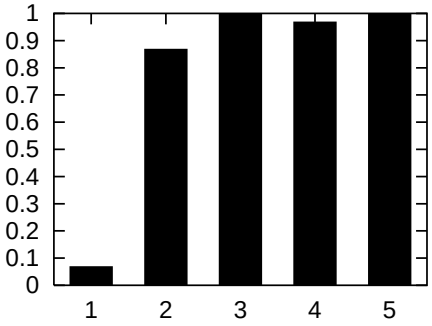

図 3: テストによる正解率．x: ランクの番号．y: テストによる正解率.

最終的に，低い正解率の問題集合のみを除去することができた．どこまで除去するかを決定する必要があるが，今回の場合，1 番低いランク集合のみを除いた場合，その総合テスト正解率は，0.96 に達することになる．

5. おわりに

各システム解の正解率を推定することにより，誤ったシステム解を持つ問題を除去する手法を提案した．その手法は EM アルゴリズムと学生の解を用いた．不十分な精度の解答生成システムであっても，この手法を用い誤りを除去することで，利用の可能性があることが確認された．今後は他の自動生成システムでも有効であるか確認したい．

参考文献

[1] Eric Brill. Some advances in transformation-based part of speech tagging. In *National Conference on Artificial Intelligence*, pages 722–727, 1994.

[2] A. O. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. In *Journal of the Royal Statistical Society*, volume 39 of *B*, pages 1–38, 1977.

[3] S. Sekine and R. Grishman. A corpus-based probabilistic grammar with only two non-terminals, 1995.