大規模言語モデルを用いたクイズ出題システムにおけるヒューマンフィードバックによる難易度自動調整の実験

須藤亘[†1]　神場知成[†1]

概要：大規模言語モデル（以下 LLM）を用いて難易度を設定した歴史分野の選択式クイズを自動生成し，ユーザ側も各問題に対して主観的な難易度を回答することで，システム側が徐々に難易度に対する認識を人間の主観とあわせて調整していくクイズ出題システムの実装を試みた．評価実験においてシステム側の調整効果は確認できなかったが，1) 問題が扱う内容自体の難しさと，選択候補を考慮したときのクイズとしての難しさの相違，2) 回答が主観に依存してしまい客観的な正解が決まらない問題が自動生成されてしまう可能性，などの課題を発見した．LLM はさまざまな分野への適用が模索されているが，このような知見は教育分野におけるパーソナライズ教育や，エンタテインメント分野におけるゲームの難易度のダイナミックな調整などさまざまな分野で活用できる可能性がある．

1. はじめに

学校教育における ICT 化が急速に進み，多くの生徒に一人一台の端末が配布される状況が整備されつつある．このような状況の中，大規模言語モデル(以下 LLM)の教育現場への導入が注目される一方，その有用性に関する議論が活発化してきている．LLM はパーソナライズされた情報を提供し，教育を効率化させられると考えられるが，生成物に誤った情報や偏見が含まれるリスクが指摘されている．

現行の研究においては，LLM が生成するコンテンツの誤情報や差別的要素を除去することに焦点を当てたものが多くみられる．しかし教育への実用化を進める上では，学習者に最適化された体験を提供することも同様に重要な課題である．本研究では「難易度」に着目し，LLM の認識を人間の感覚に近づけることを目指した．

本研究を行うにあたって，LLM を活用したクイズ出題システムを作成した．このシステムでは，回答者から提供された難易度に関するフィードバックをプロンプトとして LLM に与え，クイズの難易度が回答者の主観に合わせて調整されるかを検証した．

本研究によって，LLM の適用に関する新たな視点と方法論を提供し，教育分野におけるさらなる発展に寄与することを目指した．

2. 関連研究

2.1 RLHM（人間のフィードバックからの強化学習）

LLM を人間の価値基準に合うように人間のフィードバックを使って調整することの重要性は従来から認識され，RLHM (Reinforcement Learning from Human Feedback)と呼ばれている[1]．また，そのプロセスをオフラインで行うだけでなくオンラインで反復的に行うことの有効性についても指摘されている[2]．従来の RLHM では AI の出力が倫理的な問題，差別的な問題を引き起こさないことなどが重視される場合が多いが，AI と人間の価値基準があっていることはたとえば，パーソナライズ教育やゲームのエンタテインメント性制御などさまざまな分野で今後ますます重要になると考えられる．

2.2 教育分野における問題生成

Elkins 等[3]は教育を行うなかでクイズを出すにあたり，Bloom の分類法[4]にもとづき，学習者の記憶，理解，適用，分析，評価，創造を確認するために LLM に異なるプロンプト（記憶であれば「何を覚えていますか？」など）を与える手法を示している．Hang 等は MCQGen と呼ぶ問題生成システムの中で[5]，検索拡張生成 RAG[6]とプロンプトエンジニアリングを組み合わせることの効果を示している．

2.3 ゲームの難易度調整

ゲームにおいて，簡単すぎるか難しすぎるとプレイヤーが離脱しやすくなるため，プレイヤーの入力にもとづいて難易度をダイナミックに調整すると効果があることが報告されている[7].

以上で述べるように，一般的に LLM と人間とがさまざまなことに対して持つ感覚のレベルを合わせることは重要であり，その応用は教育，エンタテインメントなど広い範囲に及ぶと考えられる．ゲームの難易度調整の例で示されるような，難易度レベルを適切にすることでプレイヤーが離脱しにくくなるという点は，パーソナライズ教育においても重要であろう．

本論文では上記の考えにもとづき，教育的なクイズの難易度をヒューマンフィードバックと，それによるプロンプトエンジニアリングにより調整することの可能性を調査する．

†1 東洋大学　情報連携学部（INIAD）

3. 実装システム

3.1 システム構成

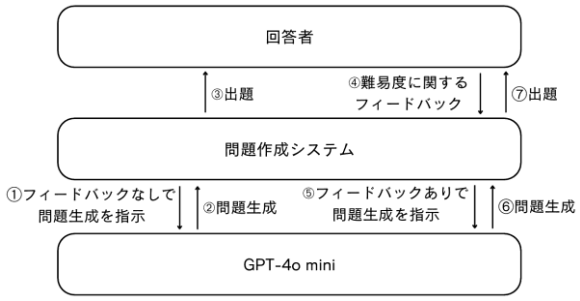

図 1. システム構成

図 1 に示した構成で，クイズ自動作成システムを作成した．これは，システムが作成した問題にユーザが回答するとともに自分が感じた難易度をフィードバックすると，システム側が徐々にそのフィードバックに基づいてユーザのレベルを考慮した問題を生成するようになるシステムである．LLM として GPT-4o mini を用い，システム本体の実装には Python の Django フレームワークを利用した．

まず，問題作成システムから GPT-4o mini の API に問い合わせを行い，ユーザからのフィードバックなしの状態で 5 題のクイズを生成する(①,②)．その際，内部的には各問題の難易度を想定して保持しておく. 生成した問題は, 1 題ずつ回答者に提示し，回答を求める(③).

その後，出題した 5 題と回答を再度提示し，ユーザにそれぞれの「難易度」を 5 段階で評価してもらい，それをフィードバックとして収集する(④). そして, 生成した問題とユーザからのフィードバックを基に，再度 GPT-4o mini の API に問い合わせて新たな問題の作成を指示する(⑤)．これにより，フィードバックを反映した新たな 5 題が作成される(⑥). この問題を再度ユーザに出題し, フィードバックを得ることを繰り返す(④～⑦)．GPT-4o mini へのプロンプトには，前回以前のユーザフィードバックも反映される．

3.2 ユーザインタフェース

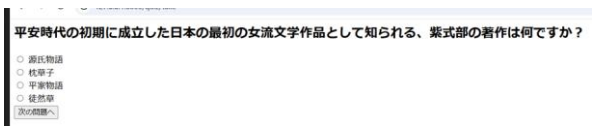

図 2. クイズ出題画面

図 2 にユーザに対する出題画面，図 3 にはユーザからのフィードバック収集画面の例を示す．図３では，問題と正解，ユーザの提出した回答を並べて表示し，5 問まとめて難易度の評価を行っている．これは 5 問まとめることで相対的な評価をしやすくなると考えたからである．

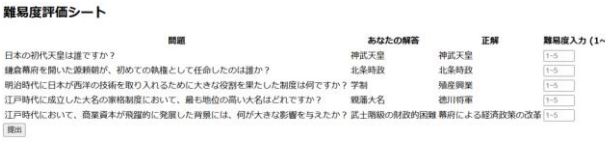

図 3. フィードバック収集画面

4. 実験

4.1 LLM へのプロンプト

実験は歴史のクイズとし，GPT-4o mini には図 4 に示すようなプロンプトを与えた．

#命令文:
あなたは{{プロの日本史教師}}です．以下の#制約条件に従い，{{最高の 4 択問題}}を作成してください．
#制約条件:
・問題のテーマは「日本史」に限定してください．
・難易度(1~5)を{difficulty}に設定し，それに応じた問題を作成してください．難易度 1 を簡単な基本問題，難易度 5 を難しい応用問題と定義します．
・以下の#フィードバックを参考に，問題の難易度や表現を調整してください．ただし，同様の問題は出題しないこと．
・歴史的な事実や人物に関しては情報の正確性を最優先してください．
・出題の根拠が不明確な選択肢や曖昧な表現を避けてください．
・以下の#出力形式に従い，{{JSON 形式}}で出力を行い，余分な説明は出力しないこと．
#フィードバック:
{feedback_text}
#出力形式:
{{
"問題": "string",
"選択肢 1": "string",
（中略）
}}

図 4. 問題作成のプロンプト

4.2 実験結果

筆者等の所属する情報系の大学 3 年生 5 名を実験協力者として実験を行った．実験は一人あたり 5 回繰り返し，フィードバックなし，フィードバック 5 問分，フィードバック 10 問分‥というように 5 問ずつ増やしていき，20 問分まで実験をした．

出力された問題，それに対してシステムが内部的に設定した難易度，それに対してユーザがフィードバックした難易度の例を図 5 に示す．

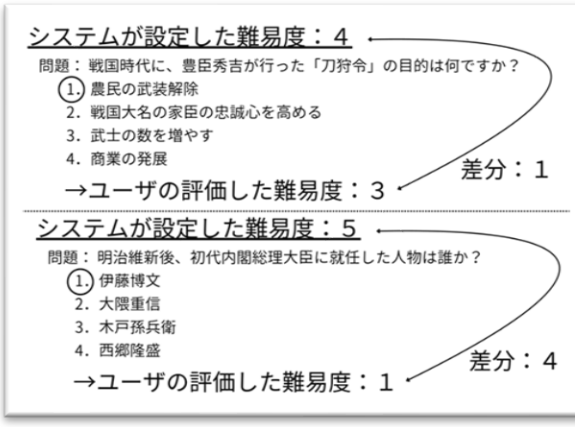

図 5. 問題例とシステム，ユーザによる難易度設定

結果として，フィードバック数ごとの，システムが設定した難易度とユーザが感じた難易度との差異は表 1 に示すように変化した．

表 1. システム設定難易度とユーザが感じた難易度の差異

	なし	5個	10個	15個	20個
ユーザA	2.00	2.00	1.80	2.40	1.00
ユーザB	1.20	2.60	2.20	0.80	0.60
ユーザC	2.20	1.00	2.00	1.80	2.20
ユーザD	0.80	0.60	2.00	1.60	1.60
ユーザE	1.20	0.40	1.00	1.20	1.40
平均	1.48	1.32	1.80	1.56	1.36

差異の変化の，全ユーザに対する平均値をグラフ化したものを図 6 に示す．

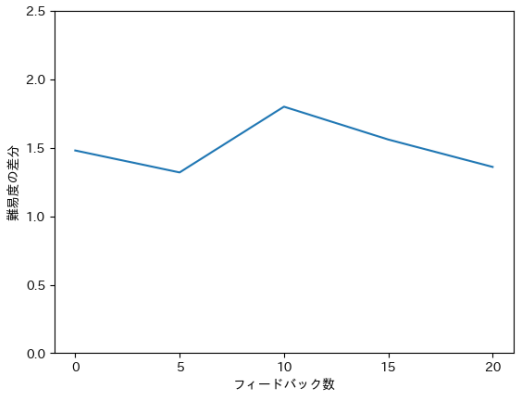

図 6 .フィードバック数ごとの難易度の差異の変化

グラフからもわかる通り，フィードバック数を増やしてもシステムが設定する難易度とユーザが感じる難易度との差異が縮小する傾向は残念ながら見られなかった．

5. 議論

システムが想定する難易度とユーザが感じる難易度との差異がフィードバックによって縮小しなかったことの原因として，次のようなものが考えられる．

1) 問題の内容自体の難しさと，クイズとしての難しさ

原因として，問題の内容自体の難しさと，クイズとしての難しさの相違があると考える．たとえば，問題の内容自体が難しかったとしても，選択肢の中で正解の選択肢が明らかに他の選択肢と異なっている場合，消去法で正解することが可能となってしまい，ユーザ視点での難易度は低く感じられることがある．一方で，問題自体の難易度が低かったとしても，選択肢が似たようなものばかりであれば，ユーザにとって難易度は高く感じられる．前述の図 5 に示した例にも，そのような傾向が見られる．また，ユーザが正解した時の難易度評価は低くなり，不正解時の難易度評価は高くなる傾向にある．今後の課題として，このような「内容の難易度」と「クイズとしての難しさ」を分離して評価する工夫が必要と考える．正解率に応じて難易度を自動設定する等も考えられるが，これは回答者群をどのように設定するかなど，別の課題が発生する．

2) 回答に主観が入り正解が決めにくい問題の自動生成

問題の曖昧な表現が難易度評価に悪影響を及ぼしていた可能性も考えられる．例えば，「江戸時代の初期，徳川家康が統一政権を確立する過程で実施した重要な戦いは何か?」というような問題に対し，「大坂の陣」と「関ヶ原の戦い」という選択肢が提示されるケースがあった．この場合，どちらを「重要な」戦いと捉えるかは主観が入る要素があり，このようなことが評価にぶれを生じさせていた．このような問題の設定を避けるためには「〜が起きたのは何年でしょう?」や「〜をしたのは誰でしょう?」というように問題文の表現をテンプレート化し，明確な問いを作成さることも考えられるが，柔軟性の高い出力がなされる生成 AI の強みが失われる懸念もあるため，曖昧性を検出して再生成を促すプロセスを導入したり，テンプレートを部分的に採用したりするなどの工夫も必要となるだろう．

その他，フィードバック数をさらに増やすことで学習材料が多様化し，精度向上が見込める可能性も考えられるが，今回のシステムではプロンプトを長くすると，「戦国時代において，織田信長が行った militarization of the country という政策とは具体的に何を指すか?」のように英語が混ざるなど生成内容が破綻することが増えた．これらは利用する LLM のバージョン等に依る部分もあると考える．

さらに，過去の問題をそのままフィードバックするのではなく，検索拡張生成（RAG）[6]の手法を用いてベクトル化して用いる等の手法も検討対象となる．

6. 結論

LLM が自動生成する選択式の歴史クイズを対象として，

回答者が感じる主観的な難易度をシステムにフィードバックすることでシステムと人間とが感じる難易度を合わせる試みを行い，調整はうまく行かなかったものの，いくつかの課題点を明らかにした．

謝辞

本研究は，東洋大学重点研究推進プログラムにより助成を受けたものです．同助成に感謝いたします．

参考文献

[1] Ouyang, Long, et al. "Training language models to follow instructions with human feedback." Advances in neural information processing systems 35 (2022): 27730-27744.
[2] Dong, Hanze, et al. "RLHF workflow: From reward modeling to online RLHF." arXiv preprint arXiv:2405.07863 (2024).
[3] Elkins, Sabina, et al. "How Teachers Can Use Large Language Models and Bloom’s Taxonomy to Create Educational Quizzes." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 38. No. 21. 2024.
[4] Krathwohl, D. R. "A Revision Bloom's Taxonomy: An Overview." Theory into Practice (2002).
[5] Hang, Ching Nam, Chee Wei Tan, and Pei-Duo Yu. "Mcqgen: A large language model-driven mcq generator for personalized learning." IEEE Access (2024).
[6] Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." Advances in Neural Information Processing Systems 33 (2020): 9459-9474.
[7] Hunicke, Robin. "The case for dynamic difficulty adjustment in games." Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology. 2005.