早押しクイズ解答システムの構築と各時点における正答率推定

杉浦 尚弥 [1]　山田 康輔 [2]　笹野 遼平 [2]　武田 浩一 [2]
[1] 名古屋大学情報学部　[2] 名古屋大学大学院情報学研究科
{sugiura.naoya.e7, yamada.kosuke.v1}@s.mail.nagoya-u.ac.jp
{sasano,takedasu}@i.nagoya-u.ac.jp

概要

本研究では、不完全な質問の入力に対して適切な解答を出力する早押しクイズ解答システムの構築に取り組む。具体的には、GPT によって質問文の後に直接解答を生成するシステムと、GPT により質問文補完を行った後に、DPR を用いた Retriever-Reader アプローチによって解答を生成するシステムを構築し評価を行う。さらに、システムの解答出力確率や関連文書を選択するスコアに基づき、質問文の各時点における正答率を推定する手法を提案する。

1　はじめに

早押しクイズとは、競技クイズのジャンルの 1 つであり、問題が読まれている途中に解答を行うことができるもので、解答する速度が重要となる性質上、不完全な文に対しての解答をすることが多いという特徴を持つ。競技クイズは対象とする知識の範囲を限定しないため、オープンドメイン質問応答タスクとして広く研究が行われているが、その主要なデータセットである Natural Questions [1] や、TriviaQA [2] は入力となる問題文が完全なものであり、これらを使用して構築されたシステム [3, 4, 5] は不完全な質問に対して解答を行うことは想定されていない。

また、早押しクイズにおいては、解答を行うだけではなく、現時点で与えられた問題文に対して予測された解答の正答らしさを考慮し、実際に解答するかどうかを判定することが重要となる。たとえば、表 1 に示す問題において「古代エジプトでは」まで読まれた場合を考えると、実際の解答である「シリウス」以外にも古代エジプトに存在した物は数多く存在することから、その時点で予測された解答は誤答である可能性が高い。一方、後半まで読まれた場合は解答はほぼ一意に定まる。このため、質問文の各時点における解答候補が実際に正解である確信度を算出することによって、より性能の高い早押しクイズ解答システムが構築できると考えられる。

表 1　早押しクイズ解答システムの出力例

完全な質問文: 古代エジプトでは「ナイルの星」と呼ばれたという、おおいぬ座の星は何でしょう? 確信度: **1.000**　解答: シリウス　**正解**
75%の質問文: 古代エジプトでは「ナイルの星」と呼ばれたという、おおいぬ座の星は何でしょう? 確信度: **1.000**　解答: シリウス　**正解**
50%の質問文: 古代エジプトでは「ナイルの星」と呼ばれたという、おおいぬ座の星は何でしょう? 確信度: **0.771**　解答: シリウス　**正解**
25%の質問文: 古代エジプトでは「ナイルの星」と呼ばれたという、おおいぬ座の星は何でしょう? 確信度: **0.125**　解答: パピルス　**不正解**

そこで本研究では、まず、不完全な質問の入力に対して適切な解答を出力する早押しクイズ解答システムを構築し、続いて、各モデルにおいて確信度の算出を行い、確信度と正答率の関係性について調査する。具体的には、GPT [6] によって質問文の後に直接解答を生成するシステムと、GPT による質問文補完を行った後、Dense Passage Retrieval (DPR) [3] を用いた Retriever-Reader アプローチによって解答を生成するシステムを構築する。また、前者のシステムでは解答生成時におけるトークン出力確率を用い、後者のシステムではモデルの出力に用いられるスコアを用いることで、確信度を算出する。表 1 に記載した確信度は、前者のスコアであるが、確信度が低い場合は不正解、確信度が高い場合は正解となる解答を出力しており、確信度が早押しクイズシステムにおいて適切に機能していることが確認できる。

2　競技クイズに対する解答システム

本節では、まず、競技クイズに対する先行研究の紹介し、その後不完全な質問文に対応する早押しクイズ解答システム、およびモデル出力に基づく確信度の算出法を提案する。

2.1 オープンドメイン質問応答モデル

競技クイズはオープンドメイン質問応答タスクとして扱われており、単一モデルで直接解答を生成するモデルと、関連文書の検索と解答箇所の抽出の 2 つの複合タスクとして解答する Retriever-Reader アプローチによる解答モデルの 2 つが主流である。

単一モデルでの解答としては GPT が代表例である。GPT とは、Transformer [7] の Decoder をベースとし、大規模なテキストコーパスを用いて、ある単語列に後続する単語を予測する学習を行った事前学習済み言語モデルである。GPT はこの性質から入力したテキストに続くテキストを生成するような言語生成関連タスクで利用でき、質問応答の場合、質問に対して解答だけを推論するような入力形式にすることで、解答を生成することができる。また、GPT では後続タスクのデータを利用したファインチューニングを行うことで、より高い性能が見込まれるため、質問応答モデル構築においてもファインチューニングすることが多い。

複合タスクでの解答としては、Retriever として DPR を用いた Retriever-Reader に基づくモデルがその代表例である。DPR は、問題文と各文書に対し異なる BERT [8] を用いる Dual-encoder 構造を持ち、BERT への入力の際、文書の先頭に挿入する特殊トークン `[CLS]` を用いて問題文、各文書の埋め込み表現を獲得する。その後それらの内積により計算された意味的な類似度を基に文書選択を行う手法である [3]。Reader では、BERT による正解を含む関連文書予測と文書内の解答箇所の抽出予測を行う。関連文書予測では、`[CLS]` トークンの位置で答えを含みそうな文書を予測する。ここで選ばれた文書に対して、解答箇所抽出予測を行い、文書中から解答となるトークン列の始点と終点を決定し出力する。

2.2 早押しクイズの解答システム

前節のシステムは完全文において有効であることは確認されているが、不完全な文の入力に対し解答することが求められる早押しクイズにおける有効性は不明である。一般的に、問題の一部のみが与えられた場合、正解となりうる解答の候補は複数存在する上、解答を行うために必要な情報が与えられていない可能性があるなど、完全文が与えられた場合と問題の性質が大きく異なる。よって早押しクイズタスクに特化した手法が必要になると考えられる。

そこで本研究では、大規模言語モデルである GPT による推論にのみに基づいて解答するシステム **GPT (推論)** と、GPT による問題文補完を行い、DPR を用いた Retriever-Reader アプローチに基づくシステム **GPT+DPR** の 2 つのシステムを構築する。GPT (推論) では、入力形式を『問題文+“/答えは「”』とすることで、括弧の先の部分を推論させ、括弧の中身を解答として出力する形式を取った。問題文と『答えは「』の間に「/」を挟むことによって、問題文の区切りを認識させ、誤って文章の補完が行われないようにした。GPT+DPR では、不完全文を GPT に入力して問題文を補完し、完全な文章を出力として得てから、それを入力として DPR を用いた Retriever-Reader に基づくモデルに解答させる形式を取った。

2.3 各モデルの確信度

本研究では、各モデルの解答に対して、各モデルが出力を生成する際に使われている内部スコアを利用して確信度を計算する。確信度はモデルの解答が正答であるかどうかの判断を行うための指標として用いる値であり、確信度が大きいほど、正答率が高くなるような性質を持つ必要がある。

GPT (推論) では、モデルが出力した解答の先頭トークンの生成確率 (以下、**生成スコア**) を確信度として利用する。GPT は文生成において、文末の n 番目までのトークンが与えられた状態で、語彙の中から最も生成スコアの高いものを $n+1$ 番目のトークンとして出力する。答えが少数のトークンで構成されることが多いクイズにおいては、先頭トークンが解答の方針のほとんどを決めることから、先頭トークンの生成スコアのみを確信度として採用した。

一方、GPT+DPR の確信度として利用可能な内部スコアとして、Reader で計算された**関連文書スコア**、**抽出スコア**、およびそれらの相加平均である**平均スコア**の 3 つを考える。Reader では各文書の `[CLS]` トークンを学習した線形層を用いてスコア化し、最も高いスコアの文書を選択するが、関連文書スコアはこのスコアである。さらに、文書選択後、選ばれた文書をトークンごとに埋め込み表現にし、別途学習した線形層を用いて始点スコアと終点スコアを計算したのち、最も始点スコアが高かったトークンから最も終点スコアが高かったトークンまでのトークン列を解答として出力するが、抽出スコアとはこの時の始点スコアと終点スコアの和である。

表 2　本研究で使用したデータセットの概要

サブセット	ソース	データ数	平均文字数
学習セット	AI 王	17,735	48.2
	みんはや	35,149	64.8
開発セット	AI 王	1,000	46.9
テストセット	AI 王	2,000	51.6

表 3　正答率検証実験の結果

モデル	25%	50%	75%	100%
GPT (推論)	0.119	0.279	0.456	0.562
GPT+DPR	0.119	0.288	0.459	0.620

3　実験

提案した早押しクイズ解答システムの正答率の検証、および、各モデルの確信度の有効性の調査を行った。なお、文字ベースで問題文を先頭 x%で切ったものを、質問文完全度 x%と表記する。正答率検証実験では、2.2 節で説明した GPT (推論) モデル、GPT+DPR モデルを、質問文完全度 25%, 50%, 75%, 100%の問題に適用し、その正答率を検証した。確信度の有効性調査実験では、各質問文完全度において、2.3 節で説明した各モデルの確信度と正解率の関係を調査することで確信度の評価を行った。

3.1　実験設定

データセット　本稿では、第 2 回 AI 王公式配布データセット (以下、AI 王)[1)]、および、クイズアプリ「みんなで早押しクイズ」で出題された問題をクローリングしたデータを用いた (以下、「みんはや」)[2)]。各データ数と平均文字数を表 2 に記載する。なお、DPR の学習は問題文と答えの他に正例文書、負例文書が必要であるため、DPR の学習は AI 王データのみを用いて行い、追加した「みんはや」データは GPT の学習にのみ使用した。

学習設定　GPT (推論) と GPT+DPR の 2 つのモデルを用いて比較を行った。GPT は rinna 社が Hugging Face [9] 上で公開しているモデル[3)]を使用した。DPR は第 2 回 AI 王で公開されたベースラインモデル[4)]を使用した。GPT (推論) は学習データを『問題文+"/答えは「○○」"』の形にフォーマットしてファインチューニングを行った。GPT+DPR においては、GPT は学習データの問題文だけを使用しファインチューニングした。いずれの場合もエポック数は 5 で学習を行った。DPR は、Retriver と Reader において事前学習済み BERT[5)]をベースとし、Retriever はバッチサイズ 128、学習率 1e-5、エポック数 5 で、Reader はバッチサイズ 8、学習率 2e-5、エポック数 3 でそれぞれ学習した。

比較手法　正答率検証実験では、GPT (推論) と GPT+DPR の 2 つのモデルを比較した。確信度の有効性調査実験では、GPT (推論) については生成スコアを、GPT+DPR については関連文書スコア、抽出スコア、平均スコアの 3 つを用いて実験を行った。

評価指標　正答率評価において、モデル出力の正誤は文字列の完全一致で評価した。ただし、表記揺れにより不正解となることを防ぐため、モデル出力と答えから『( ) [ ] ・ =』の 6 種類の記号を機械的に取り除いた後に文字列の比較を行った。

確信度の有効性調査実験では、正解率-解答率曲線を作成し、その曲線の下の面積 (AUC) により、確信度の有効性を調査した。ここで、解答率とはシステムが実際に解答を出力する割合である。確信度が閾値 α を超えるものだけを解答するものとすると、この閾値 α を変化させることで解答率を制御可能である。一方、正解率とは、システムが解答を出力したもののうち、正解だったものの割合である。α を 0 未満にした場合，これはシステム全体の正答率と一致し、α を大きくするにつれ、確信度が高いものだけが評価の対象となるので、正解率は高くなることが期待される。

3.2　実験結果: 正答率検証

GPT (推論) と GPT+DPR の質問文完全度ごと正答率を表 3 に示す。質問文完全度が下がるにつれ正答率が下がっていく結果となったが、その下がり方は質問文完全度に比例せず、100%〜75%にかけては比較的緩やかな落ち込みであった。これは、クイズの問題文の前半から中間にかけては答えを決定付ける重要な単語が多く出現するのに対し、後半に情報量の多い単語が出現するケースは少ないためだと考えられる。2 つのモデルのスコアを比較すると、質問文完全度が 100%の場合は GPT+DPR の方が高い性能となった一方で、問題文が不完全であった場合は両モデルの性能に差は確認できなかった。

1) https://sites.google.com/view/project-aio/dataset
2) https://livequiz.work/minhaya1/
3) https://huggingface.co/rinna/japanese-gpt-1b
4) https://github.com/cl-tohoku/AIO2_DPR_baseline
5) https://huggingface.co/cl-tohoku/bert-large-japanese

表 4　確信度の有効性調査実験の結果

モデル	確信度算出法	25%	50%	75%	100%
GPT (推論)	生成スコア	**0.414**	**0.636**	**0.812**	**0.859**
GPT+DPR	文書スコア	0.318	0.581	0.773	0.848
	抽出スコア	0.250	0.513	0.700	0.841
	平均スコア	0.291	0.561	0.758	0.840

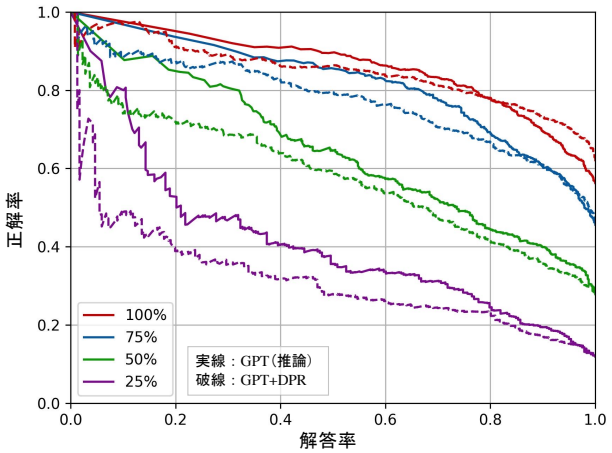

図 1　確信度の有効性調査における正解率-解答率曲線

3.3　実験結果: 確信度の有効性調査

表 4 に確信度の有効性調査実験の結果を示す。GPT+DPR に対する 3 つの確信度の中では、関連文書スコアを用いた場合が、最も高い AUC となった。また、全ての結果の中で GPT (推論) の生成スコアが最も高い AUC となった。

次に、GPT+DPR において最も AUC が高かった関連文書スコアを用いて、GPT (推論) モデルと GPT+DPR モデルの PR 曲線の比較を行った。図 1 に結果を示す。いずれの設定においても、高い確信度のもののみに解答する問題を絞り込むことで正解率は上昇することが確認でき、確信度の有効性が確認できる。GPT (推論) と GPT+DPR を比較すると、表 3 に示したとおり解答率 1.0 における正解率である正答率は、質問文完全度 100%の場合は GPT+DPR の方が高く、それ以外の場合も同等であるが、解答率 0.8 未満の領域ではいずれも GPT (推論) の方が高い正解率となった。この差は、質問文完全度が小さい場合、解答率が小さい場合に顕著であり、たとえば質問文完全度 25%、解答率 0.1 の設定においては、GPT+DPR の正解率は 0.5 程度であるのに対し、GPT (推論) の正解率は 0.8 程度と大きな差があることが確認できる。このことから、早押しクイズにおいては GPT (推論) の方が適していると言える。

表 5　GPT (推論) の質問文完全度 25%における出力例

質問 1: ごはんの上にハンバーグと目玉焼きを乗せ、グレービーソースをかけたハワイの名物料理は何でしょう?
確信度: **0.996**　解答: ロコモコ　**正解**
質問 2: 「どっどど どどうど どどうど どどう」という書き出しの一節も有名な、宮沢賢治の短編小説は何でしょう?
確信度: **0.904**　解答: 風の又三郎　**正解**
質問 3: 「英検」の正式名称は実用英語技能検定ですが、「漢検」の正式名称は何でしょう?
確信度: **0.991**　解答: 実用英語技能検定　**不正解**
質問 4: 動きが緩慢なことから名付けられた、伊豆諸島と尖閣諸島のごく一部のみに棲息する特別天然記念物の鳥は何でしょう?
確信度: **0.063**　解答: ナマケモノ　**不正解**

表 5 に、質問文完全度 25%の問題に対し、GPT (推論) を適用した場合の出力例を示す。質問 1、質問 2 に対しては、先頭 25%の質問文しか与えられていないにも関わらず、確信度スコアは 0.9 を超えており、実際に正しい解答が出力されている。一方、質問 3 に対しては、確信度スコアは高いものの、実際には誤った解答が出力されている。これは、いわゆる「ですが問題」であることを考慮に入れず予測した結果、誤った解答にも関わらず高い確信度スコアが出力されたものと考えられる。質問 4 においては、先頭 25%では解答候補が 1 つに絞り込めなかったため確信度スコアは低くなったと考えられ、実際、「アホウドリ」という正しい答えは出力できず、「ナマケモノ」という誤った解答が出力されている。

4　おわりに

本研究では、既存研究では想定されていなかった早押しクイズに解答するシステムを GPT (推論)、GPT+DPR の 2 種類構築した上で、各質問文完全度に対して正答率を検証した。また、確信度と正答率の関係性の調査、および、確信度としてモデルの内部スコアを用いる妥当性の検証を行った。実験の結果、正答率検証では、完全文において DPR が最も高いスコアを見せたものの、不完全文に対しては両モデルのスコアに差は見られなかった。確信度の有効性調査では、GPT (推論) モデルに基づく確信度スコアが低い質問文完全度においても高い有効性を示すことが確認された。今後は、本研究で対象とした確信度を用いることで、確信度の閾値を定めて解答のタイミングを決定するようなシステムや、リアルタイムな早押しクイズ解答モデルの構築を行いたい。

参考文献

[1] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. **Transactions of the Association for Computational Linguistics**, Vol. 7, , 2019.

[2] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In **ACL**, 2017.

[3] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wentau Yih. Dense passage retrieval for open-domain question answering. In **EMNLP**, pp. 6769–6781, 2020.

[4] Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. Efficient passage retrieval with hashing for open-domain question answering. In **ACL-IJCNLP**, pp. 979–986, 2021.

[5] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In **EACL**, pp. 874–880, 2021.

[6] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Open AI Technical Report, 2018.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **NIPS**, pp. 5998–6008, 2017.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **NAACL-HLT**, pp. 4171–4186, 2019.

[9] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In **EMNLP**, pp. 38–45, 2020.