多様なクイズを自動生成する手法およびその検証

小林俊介　河原大輔
早稲田大学理工学術院
{carlike787@toki., dkw@}waseda.jp

概要

クイズは、高齢者の認知機能維持や、エンターテインメントに利用されており、多様なクイズ問題の作成が必要である。しかし、人手による大量の作問にはコストがかかるため、自動生成が望ましい。本研究では、質問生成モデルの学習手法や入出力形式を変更し、多様なクイズ問題の自動生成について検証する。提案手法を用いて生成された問題は、多様な問題を生成することが確認できた。また新規の Wikipedia 記事から問題を生成できることも確認でき、システムの有効性が示された。

1　はじめに

近年、クイズがエンターテインメントのみならず、ビジネスや医療などへ応用されている。クイズを解くという行為には、思考力・判断力の向上や、記憶力・判断力の維持というメリットもあり、この観点から高齢者の認知機能維持のためにクイズが応用される例[1)]もある。しかし、クイズ問題の作問は、少量であれば人手でも負担が少ないものの、大量となると高コストになってしまう。

また、クイズを解くという行為は、自然言語処理においては質問応答タスクに該当する。しかし日本語の質問応答データセットは、英語のものと比べて量が少ないため、データの自動生成による拡張が有効な手法であると期待される。

本研究では、クイズ問題を自動で生成するシステムの構築に取り組む。クイズは問題と解答のペアで成立するが、同じ解答を導くための根拠となる知識は複数存在するため、問題で言及される内容は多様性を有すると考えられる。本研究では、多様な問題の生成を目指し、問題生成時の入出力形式を複数パターン提案し、比較検証する。通常は、根拠となる文書と想定解答を入力し、問題を出力する。しかし、入力または出力として指定する要素が少ない場合、より制限のない問題生成が可能になり、多様性に富んだ問題が出力されることが期待される。本研究では、想定解答を入力しない場合と、解答と問題を同時に出力する形式での生成を試みる。また、問題が訓練データに過適応しないよう、2 つの文章の類似度をスコアリングする BERTScore [1] を用いて、学習時の損失関数を制御する工夫を取り入れる。本研究は低コスト化を念頭に置き、大規模言語モデルを使用しない前提で実験を行った。

入出力形式を変化させて問題生成を行った結果、文書のみから問題を生成する形式で多様性が高くなった一方で、文書と解答から問題を生成することで、より矛盾のない適切な問題を生成できることが分かった。また、損失制御によって、学習時のみならず推論時にも、元から存在する問題と異なる問題を生成できることを確認した。さらに、最新の Wikipedia 記事を入力することで、学習で使用されなかった文書であっても適切な問題を生成できることを確認した。

2　関連研究

英語における質問応答データセットでは、関連文書を与えて質問に答える形式のものが多く、SQuAD [2]、TriviaQA [3]、Natural Questions [4] などがある。日本語における質問応答タスク用のデータセットには、JGLUE [5] に含まれている JSQuAD、クイズ形式の質問文で、日本語 Wikipedia を関連文書とする JAQKET [6] などがある。SQuAD と TriviaQA では訓練用の質問が 100,000 件前後、Natural Questions では 300,000 件以上用意されている。一方で JSQuAD は訓練用データで 64,000 件弱、JAQKET では評価用を合わせても 24,000 件弱と、英語データセットに対して一桁少ない。

問題の生成については、Du ら [7] が DNN による質問生成を行い、精度を改善させたことから、特にテキスト生成が可能な言語モデルによる質問生成が行われてきた。Murakhovs'ka ら [8] は、1 つの比較的

1）https://www.kaigo-antenna.jp/magazine/detail-54/

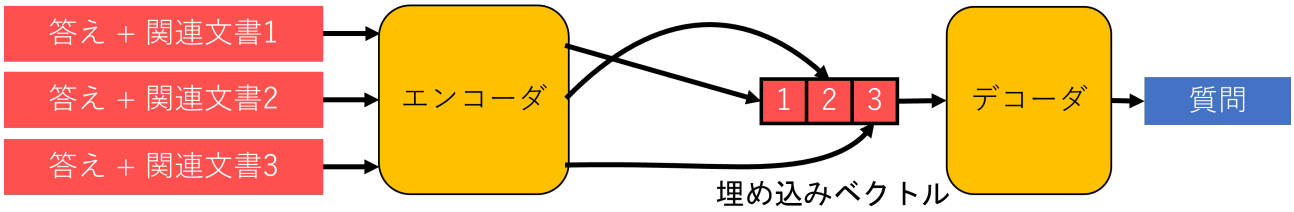

図 1　N=3 における問題生成の流れ

長い文書を入力とし、異なる解答を持つ複数の質問を生成できる MixQG を提案している。既存モデルから 10%以上の精度向上を達成しているが、入力が 1 文書であり、複数の文書を参照した質問にはなっていない。日本語においても、折原ら [9] により、日本語 T5 モデルを用いて、ニュース記事からクイズを生成する試みが行われた。既存のクイズサービスに掲載されている質問を例に、生成されたクイズが単に正答を問うだけではなく、面白みのあるクイズになっているかについて考察している。しかし、ここでも入力する記事は 1 つだけであり、また使用された訓練データ数も 220 件と少ない。また、著者ら [10] は、複数の文書を入力とした質問生成を行い、質問応答データセットの拡張によって、質問応答システムの精度を向上させている。

3　提案手法

3.1　入出力形式の変更

本研究の目標は、多様なクイズ問題の生成である。これを実現するためには、1 つの文書だけでなく複数の文書を生成時に用いることが重要であると考えられる。著者ら [10] に従い、この要素を満たす FiD を本研究での問題生成で用いるモデルとする。FiD [11] はもともと、テキスト生成モデルの 1 つである T5 [12] を用いて、Izacard ら [11] が提案した質問応答モデルで、解答生成の際に複数の文書から情報を得られるという特徴がある。本研究で使用するモデルは FiD と同一の構造であり、入出力の形式が異なるのみである。質問応答タスクでは、FiD の入力は、問題文と、読解すべき文書のタイトルおよび内容を入力するが、本研究では、問題文の代わりに解答と、解答を含む関連文書 N 個を入力する。N=3 の場合の問題生成の流れを図 1 に示す。

3.2　学習時の損失制御

この入力形式では問題のみを生成することになるが、他の形式による生成も考えられる。具体的には、関連文書のみを入力し、問題を生成することで、解答に制限されずに問題を生成できる。また、関連文書のみを入力し、問題とともに解答を同時に出力することも可能である。これらの条件では、入力時の解答に縛られない問題生成が可能であり、問題の多様性につながると期待される。

モデルの学習は、教師データを用いた教師あり学習により実施する。ただし、通常の学習を実施すると、教師データに類似した問題を生成してしまい、多様性を失う。そこで、学習時の損失関数を制御し、教師データを模倣しすぎない学習方法を提案する。具体的には、BERTScore [1] を用いて、学習時に生成された問題と、教師データの問題の類似度を計算し、損失の大きさを制御する。BERTScore は、2 つの文章の意味的類似度を算出する指標で、値が大きいほど類似している[2)]。FiD の学習では、各トークンで損失を計算し、それらの総和を最終的な損失とする。本研究では、損失を逆伝播する前に、BERTScore の算出で得られた値を用いて、高い類似度のときに損失を低減する処理を行う。具体的には、生成された問題 Q_g、教師データの問題 Q_t、言語モデルの損失 L_{LM}、BERTScore のしきい値 B_{target} を用いて、以下のように損失 L を定義する。

$$L = L_{LM} * \{B_{target} - \mathrm{BERTScore}(Q_g, Q_t)\}$$

これにより過学習を防ぎ、類似した問題の生成を防ぐことが期待される。

4　実験

4.1　実験設定

2020 年から質問応答タスクのコンペティション「AI 王」のデータセットとして JAQKET データセットが利用されている。本研究では、同コンペティションの第 2 回大会[3)]で提供されている、JAQKET

2)　BERTScore は 0 から 1 までの値をとり、全く関連のない文章同士では 0.6 程度になる。

3)　https://sites.google.com/view/project-aio/competition2

をベースとしたクイズ形式のデータセット、および日本語 Wikipedia 記事の文書集合[4]を用いた実験を行う。各問題には前処理として、Elasticsearch[5]を用い、各問題の解答を含む関連文書[6]が抽出されている。本研究ではこのデータのうち、学習用と評価用のデータセットを用いて、関連文書が 3 つ以上存在するデータを抽出し、データセットを構築した。

FiD で用いる日本語 T5 モデルは、Hugging Face Hub に存在するものを用いた[7]。問題生成で使用する文書数は、関連文書のうち 3 つとした。B_{target} は 0.9 に設定し、生成時は各入力に対し、beam search により生起確率の高い出力 7 つを取得した。

前半の実験では、入出力の形式を変更すること (3 節) で、生成される問題に変化があるか調査した。実験では入出力形式として、「文書と解答を入力し、問題を生成」、「文書のみを入力し、問題を生成」、「文書を入力し、問題と解答を生成」の 3 パターンを実験し、テストデータでの出力を問題の多様性と、生成結果の適切性から評価した。問題の多様性については、対話システムの評価指標で用いられている Distinct [13] により評価する。この指標は、生成結果に含まれるすべての n-gram のうち、ユニークなものがどれだけ存在するかという割合を計算するもので、Distinct-n とも呼ばれる。本研究では、1-gram と 2-gram を基準とし、テストデータでの出力を用いて、以下の設定における Distinct-1、2 を計算した。

1. 各入力に対し、最も生成確率が高い問題を抽出し、Distinct を計算
2. 各入力から生成された 7 つの問題で Distinct を計算し、このスコアをテストデータ全体で集計し平均を算出

また、多様性の確保には、1 つの文章だけでなく、複数の文章を用いることが重要であると考えられる。そこで無作為に抽出した 50 問を評価対象として、問題がいくつの文書を参照して生成されているかを人手評価し、問題が複数の視点を有するものになっているかについても評価した。

生成結果の適切性については、生成確率の最も高い生成結果と、テストデータに存在する問題との BERTScore を計算し評価する。

後半の実験では、BERTScore による損失関数の制御の効果を検証する。モデルは損失関数の制御を行うものと、行わないものの 2 種類で学習した。効果の検証は、以下の 3 つの手法により行った。

1. 学習中の BERTScore の変化を、2,000 ステップずつ平均したものの推移を調査
2. 1 入力から生成された 7 つの問題で Distinct を計算し、最終的な平均を算出
3. テストデータを入力した際の、全生成結果における BERTScore の平均を算出

表 1 生成確率が最大の問題における、テストセット全体での Distinct と BERTScore

モデル	Distinct-1	Distinct-2	BERTScore
文書+解答→問題	0.1818	0.5053	0.7877
文書→問題	0.1881	0.5083	0.7583
文書→問題+解答	0.1794	0.4929	0.7700

表 2 生成した問題 7 つにおける Distinct の、テストセットでの平均

モデル	Distinct-1	Distinct-2
文書+解答→問題	0.5329	0.7154
文書→問題	0.5475	0.7380
文書→問題+解答	0.5393	0.7134

4.2 実験結果と議論

4.2.1 入出力形式と問題の多様性

Distinct 及び BERTScore による評価結果を表 1 と表 2 に、人手による評価結果を表 3 に示す。

問題の多様性については、表 1 ではいずれも低い値になり、差があまり見られなかった。生成時に用いる文書は異なっていたが、問題の形式とするために「誰でしょう?」「何でしょう?」という文末を多く生成した。この影響で、ユニークな n-gram が少なくなり、スコアが低くなったと考えられる。表 2 では、文書のみから問題を生成する形式が Distinct-1、2 の双方で最高のスコアとなった。解答による制限がなくなったことで、問題生成時の条件が減り、多様な問題の生成につながったと考えられる。

表 3 の人手評価でも、文書と解答により問題を生成するパターンが、複数の文書を参考にした問題を最も多く生成した。加えて、参考文書がない問題の生成数も最も少なかった。解答を指定した問題生成であったため、他 2 つのパターンと比べ、複数の文

4) https://github.com/cl-tohoku/AIO2_DPR_baseline/blob/master/scripts/download_data.sh に記載のスクリプトでダウンロードできる。
5) https://www.elastic.co/jp/
6) 関連文書は日本語 Wikipedia 記事をパッセージの集合に分割したものである。
7) https://huggingface.co/retrieva-jp/t5-large-long

表 3　生成時に参考にした文書の数

モデル	0 文書	1 文書	2 文書以上
文書+解答→問題	4	34	12
文書→問題	12	31	7
文書→問題+解答	12	33	5

表 4　損失制御と Distinct・BERTScore

モデル	Distinct-1	Distinct-2	BERTScore
制御あり	0.5292	0.7082	0.7683
制御なし	0.5244	0.6942	0.7696

書であっても注目すべき部分が明確になったことにより、多様性につながったと考えられる。

表 1 の BERTScore による適切性の評価では、文書と解答により問題を生成するパターンで最も高いスコアとなり、文書のみから問題を生成するパターンのスコアが最も低くなった。生成時に入力・出力する情報を多くすることで、生成結果がより適切なものになると考えられる。

4.2.2　損失制御の効果

学習中の BERTScore の変化を図 2 に、Distinct と BERTScore の算出結果を表 4 に示す。

図 2 から、制御を行ったモデルでは、学習時の BERTScore の上昇が、制御を行わなかったものと比較して緩やかになっている。学習終了時点で、制御を行ったモデルと行わなかったモデルを比較すると、BERTScore の平均値は 0.008 異なっていた。また、表 4 を見ると、BERTScore や Distinct-1 では大きな差がなかったものの、Distinct-2 では、損失制御を行ったモデルが高いスコアになっており、多様な言葉を生成していることが確認できる。従って、損失制御を行った方が、既存の問題と異なる、多様な問題を生成できる傾向にあると確認できた。

5　新規 Wikipedia 記事から問題生成

AI 王のデータセットは JAQKET をベースとしているが、JAQKET は全てのデータが 2019 年時点の日本語 Wikipedia データに基づくものである。Wikipedia に掲載される記事は日々増加するため、新規に制作された記事での生成も重要である。そこで、本実験では、2023 年 6 月に初版を発行した 2 つの記事「スリーポインテッド・スター」と「マルス信州蒸溜所」を用いて、問題生成を検証した。4.1 節の条件と同様に、各記事から 3 つの文書を人手で抽出した後、人手で作成した想定解答とともに FiD へ

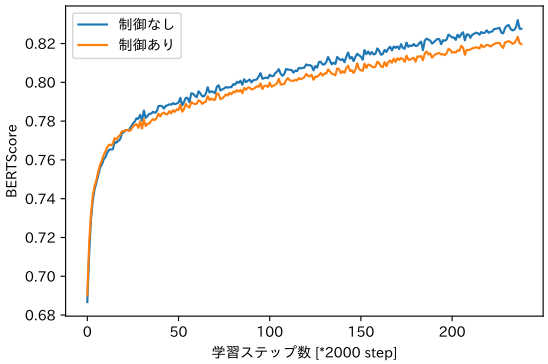

図 2　学習時の BERTScore の変化

入力し、7 つの問題を生成した。生成された問題の例を以下に示す。

1. 「スリーポインテッド・スター」から生成した例
   おなじみのスリーポインテッドスターが特徴的な、ダイムラーとベンツが合併して誕生した自動車メーカーは何でしょう? (想定解答: メルセデス・ベンツ)
2. 「マルス信州蒸溜所」から生成した例
   長野県上伊那郡宮田村にある、日本初の本格焼酎の蒸留所は何でしょう? (想定解答: マルス信州蒸留所)

これらの問題は、いずれも入力された文書の内容との矛盾が見られず、適切な問題となっている。以上から、Wikipedia の記事を用いることで、最新の情報に関する問題を生成できることが確認できた。

6　おわりに

本研究では、多様なクイズ問題の生成を試みた。これを実現するため、入出力形式の検証や、BERTScore を用いた損失関数の制御を行った。

入出力形式を変化させた場合、文書のみから問題を生成する形式で多様性が高くなった一方で、文書と解答から問題を生成することで、より複数の視点を踏まえた問題を生成できていることが分かった。また、損失制御によって、学習時のみならず推論時にも、元から存在する問題と異なる問題を生成できることを確認した。さらに、最新の Wikipedia 記事を入力することで、新たな知識であっても適切な問題を生成できることを確認した。

今後の研究では、クイズの面白さを定量化、評価する手法や、面白いクイズを生成できる文書を選択する手法について追及したい。

謝辞

本研究はキオクシア株式会社の委託研究において実施した。

参考文献

[1] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In **International Conference on Learning Representations**, 2020.

[2] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In **Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing**, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[3] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In **Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[4] 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, pp. 452–466, March 2019.

[5] Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. JGLUE: Japanese general language understanding evaluation. In **Proceedings of the Thirteenth Language Resources and Evaluation Conference**, pp. 2957–2966, Marseille, France, June 2022. European Language Resources Association.

[6] 鈴木正敏, 鈴木潤, 松田耕史, 西田京介, 井之上直也. JAQKET: クイズを題材にした日本語 QA データセットの構築. 言語処理学会第 26 回年次大会 (NLP2020) 発表論文集, pp. 237–240, Online, March 2020. 言語処理学会.

[7] Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In **Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1342–1352, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[8] Lidiya Murakhovs'ka, Chien-Sheng Wu, Philippe Laban, Tong Niu, Wenhao Liu, and Caiming Xiong. MixQG: Neural question generation with mixed answer types. In **Findings of the Association for Computational Linguistics: NAACL 2022**, pp. 1486–1497, Seattle, United States, July 2022. Association for Computational Linguistics.

[9] 折原良平, 鶴崎修功, 森岡靖太, 島田克行, 狭間智恵, 市川尚志. クイズビジネスにおける作問作業支援. 言語処理学会第 28 回年次大会 (NLP2022) 発表論文集, pp. 1401–1405, Online, March 2022. 言語処理学会.

[10] 小林俊介, 河原大輔. 複数文書の読解を要する質問の自動生成と質問応答システムへの応用. 言語処理学会第 29 回年次大会 (NLP2023) 発表論文集, pp. 2616–2621. 言語処理学会, March 2023.

[11] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**, pp. 874–880, Online, April 2021. Association for Computational Linguistics.

[12] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. **Journal of Machine Learning Research**, Vol. 21, No. 140, pp. 1–67, 2020.

[13] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 110–119, San Diego, California, June 2016. Association for Computational Linguistics.

A　新規記事による問題生成で使用した文書

5 節で行った新規 Wikipedia 記事からの問題生成において、例として示した問題の生成時に入力として用いた文書を以下に示す。

なお、以下の記事はいずれもクリエイティブ・コモンズ　表示・継承ライセンスの下で公表された、スリーポインテッド・スター及びマルス信州蒸溜所の各記事の文章をそのまま引用している。

1. スリーポインテッド・スター
   (a) スリーポインテッド・スターとは、メルセデス・ベンツの自動車などに使用される標章である。
   (b) 1926 年 6 月 28 日にダイムラー社はベンツ社と合併してダイムラー・ベンツとなり、車両の名は「メルセデス・ベンツ」が用いられ始めた。合併に際して両社の標章を融合させる形でメルセデス・ベンツ車の標章としてのスリーポインテッド・スターが完成した。
   (c) 合併後、車両のブランド名はダイムラー社の「メルセデス」とベンツ社の「ベンツ」を合わせて「メルセデス・ベンツ」となり、新たなスリーポインテッド・スターがエンブレムやフードマスコットとして用いられるようになった。
2. マルス信州蒸溜所
   (a) マルス信州蒸溜所（マルスしんしゅうじょうりゅうじょ、Mars Shinshu Distillery）は、長野県上伊那郡宮田村にあるジャパニーズ・ウイスキーの蒸留所。
   (b) マルス信州蒸溜所は 1985 年に本坊酒造によって設立された。本坊酒造は 1872 年創業の会社で、1909 年からは鹿児島県の津貫で本格焼酎の製造を手がけていた。
   (c) マルス信州蒸溜所には熟成庫が 4 つある。そのうち第 4 熟成庫は 2020 年の大改修で新設されたものである。