難易度調整機能を持つ GPT-2 に基づく読解問題自動生成手法

Difficulty Controllable GPT-2 Based Automated Question Generation for Reading Comprehension Test

鈴木 彩香 *1，宇都雅輝 *1
Ayaka Suzuki*1, Masaki Uto*1
*1 電気通信大学
*1The University of Electro-Communications
Email: {suzuki_ayaka, uto}@ai.lab.uec.ac.jp

あらまし： 近年，任意の長文に関連する読解問題を深層学習を用いて自動生成する読解問題自動生成手法が注目されている．最先端手法では，与えられた長文と整合性がある自然な問題文を生成できるが，生成される問題の難易度は考慮できなかった．そこで本研究では，任意の難易度の読解問題を自動生成する手法を開発する．具体的には，深層学習言語モデル GPT-2 を用いた読解問題自動生成手法に対して，項目反応理論を利用して推定される問題難易度を組み込んだ入力データを与えることで，所望の難易度に合わせた問題を生成できる技術を提案する．

キーワード： 読解問題，問題生成，深層学習，言語モデル，項目反応理論，言語生成

1　はじめに

読解問題自動生成とは，与えられた長文からそれに関連する問題を自動生成する技術であり，教育分野において読解力を育成・評価するアプローチの一つとして活用が期待されている．読解問題自動生成手法として，従来は人手で設計したテンプレートを利用するルールベースの手法が主流であったが，近年では深層学習を用いた手法が多数提案されている [1][2]．最先端の深層学習ベースの手法は，人手でのテンプレート作成を行うことなく，柔軟で高品質な問題生成を実現している．

一方，既存の問題自動生成手法では，読解対象の長文（以降では「提示文」と呼ぶ）と整合性があり，文法的に正しい問題を生成することを目標としており，生成される問題の難易度などの特性は考慮されていない．しかし，読解力の効率的な育成支援のためには，学生の読解力のレベルに合わせた問題生成が必要と考えられる．

そこで，本研究では，任意の難易度の問題を自動生成する手法を提案する．具体的には，事前学習言語モデルの一つである GPT-2（Generative Pre-trained Transformer 2）[3] を用いた深層学習ベースの読解問題自動生成手法に対して，項目反応理論（Item response theory : IRT）を利用して推定される各問題の難易度を組み込んだ入力データを与えることで，所望の難易度に合わせた問題を生成することを目指す．

2　提案手法

上述した通り，本研究では基礎モデルに GPT-2 を用いる．GPT-2 は，15 億以上のパラメータを持つ Transformer ベースの大規模深層学習モデルを，800 万以上の文書データで教師なし学習することにより汎用的な言語構造を獲得させた事前学習言語モデルであり，様々な言語生成タスクで高性能を達成している．そこで，本研究では，GPT-2 を用いた問題自動生成手法を，問題の難易度を調整できるように拡張する．

2.1　難易度を含んだデータセットの作成

提案手法では，問題の文章情報に加えて，それらの問題の難易度も訓練データとして使用する．各問題の難易度は，IRT を用いて以下の手順で推定する．

1. **各問題に対する正誤反応データの収集：**訓練データ中の各問題を実際に出題して正誤反応データを収集する．ただし，本研究では人間の解答者を QA（Question Answering）システムで代用する．
2. **IRT を用いた難易度推定：**最も単純な IRT モデルである次式のラッシュモデルを利用して，正誤反応データから各問題の難易度を推定する．

$$p = \frac{\exp(\theta_r - b^{(s)})}{1 + \exp(\theta_r - b^{(s)})} \tag{1}$$

ここで，$b^{(s)}$ は s 番目の問題の難易度，θ_r は r 番目の解答者の能力値を表すパラメータである．

3. **推定された難易度を含んだデータセットの作成：**IRT で推定された難易度を用いて，提案手法のための訓練データセット $\boldsymbol{C}$ を作成する．データセット $\boldsymbol{C}$ は，提示文 $\boldsymbol{w}^{(s)}$，答え $\boldsymbol{a}^{(s)}$，問題 $\boldsymbol{q}^{(s)}$，難易度 $b^{(s)}$ の集合として，以下のように表記できる．

$$\boldsymbol{C} = \{(\boldsymbol{w}^{(s)}, \boldsymbol{a}^{(s)}, \boldsymbol{q}^{(s)}, b^{(s)}) \,|\, s \in \{1, \cdots, S\}\} \tag{2}$$

ここで，S はデータ数を表す．また，提示文 $\boldsymbol{w}^{(s)}$，答え $\boldsymbol{a}^{(s)}$，問題 $\boldsymbol{q}^{(s)}$ は単語の系列として，$\boldsymbol{w}^{(s)} = \{w_n^{(s)} \,|\, n \in \{1, \cdots, N^{(s)}\}\}$，$\boldsymbol{a}^{(s)} = \{a_m^{(s)} \,|\, m \in \{1, \cdots, M^{(s)}\}\}$，$\boldsymbol{q}^{(s)} = \{q_o^{(s)} \,|\, o \in \{1, \cdots, O^{(s)}\}\}$ と定義する．$N^{(s)}$，$M^{(s)}$，$O^{(s)}$ はそれぞれ $\boldsymbol{w}^{(s)}$，$\boldsymbol{a}^{(s)}$，$\boldsymbol{q}^{(s)}$ 内の単語数を表し，$w_n^{(s)}$，$a_m^{(s)}$，$q_o^{(s)}$ はそれぞれ提示文，答え，問題文の中の添字に対応する位置の単語を表す．なお，以降では特別に明記しない場合には，$\boldsymbol{w}$, $\boldsymbol{a}$，$\boldsymbol{q}$，b は，任意の s に対する $\boldsymbol{w}^{(s)}$, $\boldsymbol{a}^{(s)}$，$\boldsymbol{q}^{(s)}$，$b^{(s)}$ を表すこととする．

このデータを用いて，提案手法では，1）提示文と指定した難易度から答えを生成するモデルと，2）生成され

た答えと提示文，および指定した難易度から問題を生成するモデル，の 2 段階モデルで問題生成を実現する．以降で各モデルの詳細を説明する．

2.2　難易度調整可能な答え生成モデル

まず，提示文と指定した難易度に基づいて答えを生成する提案モデルを説明する．このモデルでは，提示文 $\boldsymbol{w}$ と答え $\boldsymbol{a}$，難易度 b をいくつかの特殊トークンで連結した以下の構造のデータを学習データとする．

$$<\mathrm{BOS}>\ b\ <\mathrm{QU}>\ \boldsymbol{w}\ <\mathrm{G}>\ \boldsymbol{a}\ <\mathrm{EOS}> \quad (3)$$

ここで，<QU>は提示文の始まりを，<G>は生成対象文の開始を，<BOS>と<EOS>はデータの開始と終了を表す特殊トークンである．

モデル学習は以下の損失関数を用いて行う．

$$L_{\boldsymbol{a}} = -\sum_{s=1}^{S}\sum_{m=1}^{M^{(s)}} \log P(a_m^{(s)} \mid a_1^{(s)}, \cdots, a_{m-1}^{(s)}, \boldsymbol{w}^{(s)}, b^{(s)})$$

学習されたモデルを用いた答えの生成は，<G>までのデータを入力として与え，次式の尤度 $P(\boldsymbol{a})$ を最大化する答えの文 $\hat{\boldsymbol{a}} = \arg\max_{\boldsymbol{a}} P(\boldsymbol{a})$ を出力することで行う．

$$P(\boldsymbol{a}) = \prod_{m=1}^{M} P(a_m \mid a_1, \cdots, a_{m-1}, \boldsymbol{w}, b)$$

2.3　難易度調整可能な問題生成モデル

次に，生成された答えと提示文，および指定した難易度から問題を生成する提案モデルを説明する．このモデルでは，提示文 $\boldsymbol{w}$ と答え $\boldsymbol{a}$，問題 $\boldsymbol{q}$，難易度 b を特殊トークンで連結した以下のデータを学習に用いる．

$$<\mathrm{BOS}>\ b\ <\mathrm{QU}>\ \boldsymbol{w}\ <\mathrm{AN}>\ \boldsymbol{a}\ <\mathrm{G}>\ \boldsymbol{q}\ <\mathrm{EOS}> \quad (4)$$

ここで，<AN>は答えの開始を表す特殊トークンである．

モデル学習に用いる損失関数は以下の通りである．

$$L_{\boldsymbol{q}} = -\sum_{s=1}^{S}\sum_{o=1}^{O^{(s)}} \log P(q_o^{(s)} \mid q_1^{(s)}, \cdots, q_{o-1}^{(s)}, \boldsymbol{w}^{(s)}, \boldsymbol{a}^{(s)}, b^{(s)})$$

学習されたモデルを用いた問題生成は，<G>までのデータを入力として与え，次式の尤度 $P(\boldsymbol{q})$ を最大化する問題文 $\hat{\boldsymbol{q}} = \arg\max_{\boldsymbol{q}} P(\boldsymbol{q})$ を出力することで行う．

$$P(\boldsymbol{q}) = \prod_{o=1}^{O} P(q_o \mid q_1, \cdots, q_{o-1}, \boldsymbol{w}, \boldsymbol{a}, b)$$

3　提案手法の有効性評価実験

ここでは，提案手法の有効性を評価する実験について説明する．本実験の手順は次の通りである．1）質問応答・問題生成タスクで広く利用される SQuAD データセットの訓練データを用いて，精度に差がある 5 つの QA システムを構築した．2）5 つの QA システムに SQuAD のテストデータ中の各問題を解答させ，正誤反応データを収集した．3）得られた正誤反応データを用いて，式 (1) のラッシュモデルで各問題の難易度推定を行なった．得られた難易度推定値は，-3.96，-1.82，-0.26，0.88，2.00，3.60 のいずれかの値となり，値が小さいほど簡単な問題であることを意味する．4）得られた難易度推定値と SQuAD のテストデータの情報を使用し，式 (3) と式 (4) の通りに提案手法で使用するデータを作成した．5）手順 4 で作成したデータを 90% と 10% に分割し，90% のデータで提案手法のモデル学習（ファインチューニング）を行ない，残り 10% のデータで所望の難易度に応じた生成が行えたかを「生成された答えの難易度別平均単語数」と「生成された問題の難易度別正答率」の 2 つの観点で評価した．なお，正答率の評価には 2 つの QA システムを使用し，2 つの QA システムのどちらか 1 つでも正解した問題を正答として扱った．

表 1　答えの難易度別平均単語数と問題の難易度別正答率

指定した難易度	答えの平均単語数	問題への正答率
-3.96	2.56	47.59
-1.82	2.78	47.32
-0.26	3.55	38.25
0.88	3.35	34.29
2.00	3.82	22.89
3.60	4.72	20.03

表 2　生成された問題と答えの例

指定した難易度	-3.96
生成された問題	Who submits the bill to the monarch for royal assent?
生成された答え	Presiding Officer
指定した難易度	3.60
生成された問題	Why is Islamism controversial?
生成された答え	not just because it posits a political role for Islam but also because its supporters believe their views merely reflect

結果を表 1 に示す．表から，指定する難易度が高いほど，生成された答えの平均単語数が増加し，生成された問題の正答率が減少する傾向が確認できる．このことから，提案手法により生成した答えや問題が，指定した難易度を適切に反映していることがわかる．

表 2 に生成された問題と答えの例を示す．表から，低い難易度を指定した場合には，単一の用語が答えとなるような簡単な問題が生成されたのに対し，高い難易度を指定した場合には，理由を記述させるようなより難しい問題が生成されたことがわかる．

4　まとめと今後の課題

本研究では，任意の難易度の問題を自動生成する手法を提案し，実験から提案手法の有効性を示した．今後は，QA システムではなく人間を対象にしたデータ収集と評価実験を行っていきたい．

参考文献

[1] Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pp. 1342–1352, Vancouver, Canada, 2017.

[2] Ying-Hong Chan and Yao-Chung Fan. A recurrent BERT-based model for question generation. In *Proc. Workshop on Machine Reading for Question Answering*, pp. 154–162. Association for Computational Linguistics, 2019.

[3] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.