Vol.2019-NL-241 No.15
2019/8/29

クイズ解答タスクにおける 大規模ラベルなしコーパスの利用: 言語モデルとデータ拡張

鈴木 正敏[1,a)]　松田 耕史[2,1,b)]　大内 啓樹[2,1,c)]　鈴木 潤[1,2,d)]　乾 健太郎[1,2,e)]

概要：Quizbowl は，複数の文からなるクイズ問題の入力に対して，正解となるエンティティを予測する質問応答タスクである．Quizbowl は超多クラス（数万クラス以上）の分類問題と見なすことができるが，その場合，訓練データの規模が限定的であるために few-shot 学習の問題が生じる．すなわち，訓練データにわずかな回数しか出現しないエンティティであっても，テスト時には正しく答えなければならない．この問題に対処するため，本研究では，1) 大規模コーパスで事前訓練された言語モデルの利用と，2) Wikipedia を利用したデータ拡張を組み合わせたクイズ解答の手法を提案する．具体的には，1) 汎用言語モデル BERT の複数の層が出力する分散表現を用いて，クイズ問題から正解エンティティへのマッピングを学習する．さらに，2) Wikipedia の記事の性質を利用して，テキスト-エンティティ対のデータを大量に生成し，擬似クイズ問題として訓練データに追加する．これらモデルとデータ両方向の拡張により，Quizbowl のクイズ解答の性能が大幅に向上することを実験的に示す．

キーワード：質問応答システム，言語モデル，データ拡張

1. はじめに

クイズはファクトイド型質問応答の一種である．Jeopardy![*1]や Who Wants to be a Millionaire[*2]に代表されるように，人間同士が知識を競い合う活動として多くの人々の関心を集めてきた．クイズの問題に正しく解答するには，豊富な知識の保持，関連する既存知識の想起，既存知識を用いた推論などのスキルが要求される．これらのスキルを機械にどのように習得させるかという問題は人工知能研究の主要なテーマであり，多くの研究が行われてきた [1]．

本研究では，クイズ問題を用いた質問応答タスクの Quizbowl に取り組む．Quizbowl は，早押しクイズの問題文の入力に対して，正解となるエンティティを予測する質問応答タスクである．これは，問題文の入力に対して，正解となるエンティティを予測する分類問題とみなすことができる．ただし，正解となり得るエンティティは膨大な数が存在する．例えば英語版の Wikipedia には本稿執筆時でおよそ 590 万記事が存在する．これらを正解エンティティ候補集合とみなすと 590 万クラスの分類問題となる．すなわち，Quizbowl タスクは本質的に**超多クラスの分類問題**となる．

超多クラス分類問題で避けることのできない問題に，**few-shot 学習**の問題がある．超多クラス分類問題では，訓練データセットの量に対して，正解となり得るクラスの数が相対的に大きくなるため，一部のクラスは訓練データセットにわずかな回数しか出現しないという問題が生じる（2.2 節）．このような訓練データ中の低頻度クラスの事例に対しても，テスト時には正しい分類が要求される．

Quizbowl タスクにおける few-shot 学習の問題に対処するため，本研究では大規模ラベルなしコーパスを利用した 2 つの方法を提案する．1 つは，大規模なコーパスで事前訓練された汎用言語モデルを利用することである．最近の研究で，ELMo[2] や BERT[3] といった汎用的な言語モデルが，自然言語処理の各種タスクで高い性能を示すことが報告されている．これらの言語モデルは，タスクに依存しない大規模なラベルなしコーパスを用いて事前訓練済みであるため，訓練データの不足に起因する few-shot 学習においても有効であることが期待できる．本稿では，BERT の

[1] 東北大学
[2] 理化学研究所
[a)] m.suzuki@ecei.tohoku.ac.jp
[b)] koji.matsuda@riken.jp
[c)] hiroki.ouchi@riken.jp
[d)] jun.suzuki@ecei.tohoku.ac.jp
[e)] inui@ecei.tohoku.ac.jp
[*1] https://www.jeopardy.com/
[*2] https://millionairetv.dadt.com/

Question: The protagonist of one of this man's works gives a jar to his friend instead of repaying a loan, and later dies after embezzling money in an attempt to buy out a courtesan's contract. Another of his works sees a man nicknamed “Hard Luck” stab a tobacco merchant in the head after drinking in a teahouse with Yojibei and Azuma. The title character of another work by this man is aided by the generals Kanki and Go Sankei in defeating the Manchu forces under Ri Toten. This author of The Courier for Hell and The Uprooted Pine also wrote a play in which the oil merchant Kuheiji scams Tokubei out of his dowry, after which Tokubei and Ohatsu kill themselves. For 10 points, name this author of such bunraku plays as The Battles of Coxinga and The Love Suicides at Sonezaki.
Answer: Chikamatsu Monzaemon

図 1 Quizbowl の問題例

持つ複数の層に由来する分散表現を用いて，クイズ問題から正解エンティティへのマッピングを学習する (3 節).

もう 1 つは，大規模なコーパスである Wikipedia を用いてクイズの訓練データを拡張することである．自然言語処理の研究で，解きたいタスクのデータセット以外の言語資源から擬似的に訓練データを作成し，モデルの訓練に用いるという，訓練データ拡張の手法は，主に機械翻訳タスクで盛んに研究されている [4], [5]．しかし，Quizbowl のような質問応答タスクにおける多クラス分類において，訓練データ拡張の有効性についての十分な研究はなされていない． 本研究で取り組む Quizbowl タスクでは，問題テキストから正解エンティティへのマッピングを学習することが要求される．したがって，テキスト-エンティティ対のデータを擬似的なクイズ問題の訓練データとして大量に用意できれば，システムの解答性能の向上に寄与できる可能性がある．本稿では，Wikipedia 記事中の各文と記事タイトル（エンティティ）を紐付けたテキスト-エンティティ対のデータを大量に生成し，擬似クイズ問題として訓練データに追加する（4 節）.

以上のような，大規模ラベルなしコーパスを利用したモデルとデータ両方向の拡張によって，超多クラス分類問題としての Quizbowl タスクにおける性能向上を実験的に示す．本研究の貢献は，以下の通りである．

- クイズ解答タスクにおける多クラス分類問題において，汎用言語モデル BERT の有効性を実証．さらに，問題文が短く解答の手がかりが少ない状況下で，BERT の持つ複数の層の利用による性能向上の発見．
- Wikipedia を利用した訓練データ拡張による解答性能の大幅な向上.
- few-shot（訓練データにおいて低頻度の）エンティティに対する性能分析.
- 問題の難易度ごとの性能分析と，クイズ問題の pyramidality 性（2.2 を参照）の実証.

表 1 QANTA データセットの統計

	問題数	問題文数	正解の異なり数
train	112,927	587,895	25,969
dev	2,216	14,269	1,946
test	4,104	26,679	3,264

2. Quizbowl タスク

Quizbowl は，主に英米圏の学生の間で行われているクイズの大会である．Quizbowl で出題されるクイズ問題は，出題者が問題を読み上げる途中であっても解答者はブザーを押して解答することができる「早押しクイズ」である．出題される問題のジャンルは，文学，歴史，科学，地理など，アカデミックな分野を中心に幅広い.

図 1 に Quizbowl のクイズ問題の例を示す．1 つのクイズ問題は，問題 (Question) と正解 (Answer) のペアからなる．問題は，複数の文で構成されており，正解のほとんどは固有表現である.

2.1 QANTA データセット

QANTA[6] は，Quizbowl のクイズ問題を利用した質問応答のデータセットである．本稿執筆時点での最新版である QANTA 2018 データセットは, Quizbowl で過去に使用されたおよそ 12 万問の問題 x_q と正解 c のペア (x_q, c) からなるデータセットである．問題 x_q は複数の文 $x_{q,1}, x_{q,2}, \cdots, x_{q,m_q}$ からなる.[*3] 正解 c は，Wikipedia 記事のタイトルで示される固有表現に限定されている．QANTA 2018 データセットの各種統計量を表 1 に示す．1 問のクイズ問題は，平均して 5 文から 6 文で構成されている.

QANTA 2018 データセットでは，すべてのクイズ問題の答えが Wikipedia 記事のタイトルに限定されている．そのため，QANTA は，問題テキストの入力に対して，Wikipedia の記事タイトルの集合（大きさは数万〜数百万となる）を解の空間とした，超多クラスの分類問題として解くことを前提としたデータセットとなっている．さらに，QANTA 2018 データセットには，クイズ問題に紐づけられたすべての Wikipedia 記事の本文ファイルも補助的なデータとして含まれている．本研究では，この Wikipedia 記事のデータを，クイズ問題の訓練データを拡張するために用いる（4 節）.

2.2 多クラス分類問題としての Quizbowl

Quizbowl のクイズ問題の重要な性質として，次の 2 点が挙げられる.

1 点目は，pyramildality と呼ばれている性質である [7]. Quizbowl の問題は，正解となっているエンティティにつ

[*3] 文分割 (文境界) 情報もデータセットに含まれている.

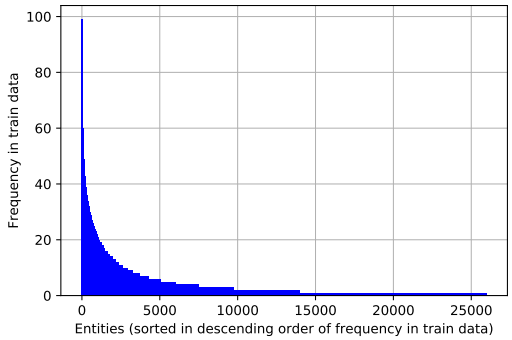

図 2　train データにおける正解エンティティの出現頻度の分布

いてより詳しく知っている解答者がより早く正答できるよう，より難しいヒントが先に，より易しいヒントが後に現れるように作られている．例えば，図 1 の問題の正解は「近松門左衛門」である．問題の 1 文目では，近松によるある作品のワンシーンについて書かれているが，このヒントだけで正解を特定するのは難しい．一方，最終文では近松の代表作のタイトルが挙げられており，正解の人物を特定しやすくなっている．この pyramidality の性質によって，問題文 $x_{q,i}$ が問題 x_q の何文目であるかの情報が，問題文 $x_{q,i}$ の難易度の大まかな指標となる．さらに，Quizbowl では，任意の問題 x_q のどの問題文 $x_{q,i}$ を 1 つ取り出しても，$x_{q,i}$ の情報だけで正解 c が一意に定まるように作問されている．したがって，問題を 1 文ずつに区切ってシステムに入力し答えさせることで，システムの解答性能を問題の難易度ごとに分析することが可能となる．

2 点目は，訓練データの数に対して正解エンティティの異なり数（種類数）が多く，さらに正解エンティティの出現頻度の分布が大きく偏っている点である．QANTA の train データセットには 25,969 種類のエンティティが正解として出現するが（表 1），その出現頻度の分布は大きく偏っている．train データにおける正解エンティティの出現頻度の分布を図 2 に示す．train データで最も出現頻度の高い正解エンティティは「Japan」であり，合計 99 回 train データに出現する．その一方で，train データに 1 回しか出現しないエンティティは 12,024 個存在する．よって，Quizbowl の問題を多クラス分類問題として解くには，訓練データにわずかな回数しか出現しない正解エンティティであってもテスト時には正しく答えなければならない，いわゆる few-shot 学習の問題が生じる．

3. クイズ解答モデル TriviaBERT

2.2 節で述べたように，Quizbowl を多クラス分類問題として解く QANTA の問題設定では few-shot 学習の問題が生じる．この問題に対処する方法の 1 つとして，本研究では，大規模コーパスで事前訓練された汎用言語モデルである BERT [3] を利用することを考える．

BERT は大規模なラベルなしコーパスで訓練された汎用言語モデルである．BERT を極性判定や文書分類などの分類タスクに利用するには，入力文に対して BERT が出力する文ベクトル（具体的には，入力文の文頭に付加される [CLS] トークンに対して出力するベクトル）を，分類タスクのための分類器への入力とする．

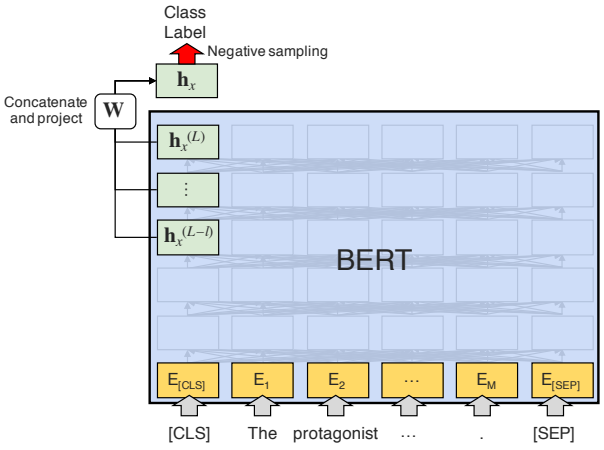

図 3　TriviaBERT モデル

BERT を QANTA のような超多クラス分類問題に適用するには，次の 2 点について考慮する必要がある．1 つは，クラス数が大きいために，通常の softmax 関数を利用したスコア計算は，計算にかかる時間の観点からは困難になるという点である．もう 1 つは，すべてのクラスの重みベクトルをモデルに保持させることは，計算機のメモリ使用量の観点からは難しいという点である．

本研究では，これらの問題に対処するクイズ解答モデル TriviaBERT を提案する．TriviaBERT は，BERT が出力する文ベクトルを利用して超多クラス分類問題を解くモデルである．TriviaBERT の基本構造を図 3 に示す．TriviaBERT は，通常の BERT のモデルにいくつかの変更を加え，超多クラス分類問題に対応させたものである．以降の節ではこのモデルの変更点について説明する．

3.1 計算効率の向上：負例サンプリング

通常，BERT で多クラス分類問題を解くときには，入力文 x に対し BERT の最終層が出力する文ベクトル（[CLS] トークンに対して出力するベクトル）$\mathbf{h}_x$ と出力クラス c の重みベクトル $\mathbf{w}_c$ との内積計算により，クラス $c \in C$ が正解クラスである確率 $P(c|x)$ を求める．

$$P(c|x) = \frac{\exp(\mathbf{h}_x^\top \mathbf{w}_c)}{\sum_{c' \in C} \exp(\mathbf{h}_x^\top \mathbf{w}_{c'})} \quad (1)$$

この計算を超多クラス分類問題に適用しようとすると，(1) 式の分母の計算に時間がかかり，サンプルあたりの計算が遅くなってしまう．この問題に対処するため，モデルの訓練時には，$P(c|x)$ の負の対数尤度を負例サンプリング [8] により近似した以下の値 ℓ を求め，これを最小化の目的関数とする．

表 2　Wikipedia から作成された拡張データの統計

	エンティティ数	文数（エンティティ平均）
Wiki-all	25,765	3,417,066 (132.62)
Wiki-lead	25,765	93,076 (3.62)

$$\ell = -\log \sigma\left(\mathbf{h}_x^\top \mathbf{w}_c\right) - \sum_{c' \in C'} \log \sigma\left(-\mathbf{h}_x^\top \mathbf{w}_{c'}\right) \qquad (2)$$

ただし，C' は C の部分集合で，一様分布からサンプルされた K 個の要素からなる．

3.2　メモリ使用量の削減: 文ベクトルの変換

(1) 式および (2) 式では，クラス $c \in C$ の数だけ，重みベクトル $\mathbf{w}_c$ を用意する必要がある．しかし，超多クラス分類問題では，BERT が出力するベクトルと同じ次元 d の重みベクトルをすべて計算機のメモリ上に展開することは，クラスの数が大きければ大きいほど困難になる．そこで，TriviaBERT では，クラスの重みベクトル $\mathbf{w}_c$ の次元を d' $(d' \leq d)$ とし，演算時には BERT が出力する文ベクトル $\mathbf{h}_x^{(L)} \in \mathbb{R}^d$ を線形変換して d' 次元としてから $\mathbf{w}_c$ との内積計算を行う．

$$\mathbf{h}_x = \mathbf{h}_x^{(L)} \mathbf{W} \qquad (3)$$

ただし，$\mathbf{W} \in \mathbb{R}^{d \times d'}$ は変換行列である．

ところで，BERT で分類問題を解く場合，BERT の最終層（出力側に最も近い層）が出力するベクトルのみを分類器の入力とする場合が多い．一方，最近の研究では，BERT は層によって異なる抽象度の情報を捉えていることが示唆されている [9]．そこで本研究では，BERT 複数の層が出力する情報を利用するため，BERT の出力に近い方から数えて l 番目までの層が出力する文ベクトルを連結し線形変換を適用したものを，クラスの重みベクトルとの内積計算に用いる．すなわち，(1) 式および (2) 式において，

$$\mathbf{h}_x = [\mathbf{h}_x^{(L)}; \mathbf{h}_x^{(L-1)}; \mathbf{h}_x^{(L-2)}; \cdots; \mathbf{h}_x^{(L-l+1)}]\mathbf{W} \qquad (4)$$

とする．ただし，$\mathbf{h}_x^{(i)}$ は BERT の i 番目の層が出力する文ベクトルを，$[\cdot;\cdot]$ はベクトル同士の連結をそれぞれ表す．$\mathbf{W} \in \mathbb{R}^{ld \times d'}$ と $\mathbf{w}_c \in \mathbb{R}^{d'}$ $(c \in C)$ は，TriviaBERT モデルの訓練可能なパラメータである．

4.　Wikipedia を利用したデータ拡張

3 節で述べた提案モデルの元となっている BERT は，大規模なコーパスで事前訓練を行った後に個々のタスクで fine tuning を行うことで，fine tuning に用いるデータセットが少量であっても性能向上を実現できるモデルである．本研究では，BERT の事前訓練の着想に基づき，Wikipedia を用いて作成した拡張データで TriviaBERT モデルを事前訓練することで，少量のクイズ問題のデータに対しても解答性能を向上させる手法を提案する．

クイズ問題を多クラス分類問題として解くことは，問題（テキスト）から正解（エンティティ）へのマッピングを学習する問題であると見なすことができる．例えば，**「ローマを首都とする国は？」「ティラミスはどこの国の洋菓子？」「レオナルド・ダ・ヴィンチが生まれた国は？」**などの文はすべて「イタリア」が正解の問題文である．システムはこれらの文から正解の「イタリア」への意味的なマッピングが学習できれば，これらの問題に正解できるようになると考えられる．しかし，表 2 にも示したように，各正解エンティティに対応する問題文は訓練データに少数しか含まれていない（約半数のエンティティは 1 つの問題文としか対応しない）ため，問題文とエンティティの適切なマッピングを学習するには不十分である．そこで，クイズ問題とは別にテキスト-エンティティ対のデータを大量に用意できれば，それらをクイズ問題の擬似データとみなしてモデルの訓練に利用することで，クイズ問題の訓練データにはわずかな回数しか出現しないエンティティであっても正しく答えられるようにモデルを訓練できる可能性がある．

本研究では，Wikipedia の記事の性質を利用して，テキスト-エンティティ対のデータを大量に生成し，訓練データに追加することを提案する．通常，Wikipedia 記事の本文は，記事タイトルとなっているエンティティについての文章である．一方で，Quizbowl のクイズ問題は，問題文が正解エンティティについての文章となっている．すなわち，Wikipedia の記事と Quizbowl の問題には，テキストがエンティティについての文章になっている，という共通する性質がある．

この共通点に着目し，Wikipedia 記事を利用した，QANTA の訓練データの拡張を以下の方法により行った．最初に，QANTA の train データに出現するすべての正解エンティティ C を抽出した．次に，各正解エンティティ $c \in C$ について，エンティティに紐づいた Wikipedia 記事の本文テキスト x_c を文分割して $x_{c,1}, x_{c,2}, \cdots, x_{c,m_c}$ とし，1 つのエンティティにつき複数の文-記事タイトル対 $(x_{c,1}, c), (x_{c,2}, c), \cdots, (x_{c,m_c}, c)$ を得た．すべてのエンティティ $c \in C$ について得られた文-記事タイトル対を，テキスト-エンティティ対のデータとみなし，訓練データの拡張データとした．なお，Wikipedia 記事の本文テキストのデータは，QANTA のクイズ問題とともに配布されているものを使用した．

拡張データは，Wikipedia 記事の本文全文を x_c として用いた **Wiki-all** と，本文の最初の段落のみを x_c として用いた **Wiki-lead** の 2 種類を作成した．拡張データの各種統計量を表 2 に示す．拡張データ作成の元データである QANTA の Wikipedia 記事のデータには一部のエンティティに対する本文データが欠落しているため，拡張データのエンティティの数は，QANTA の train データに出現するエンティティの種類数（表 1）よりも少なくなっている．

表 3　Wikipedia から作成された拡張データの例

	テキスト x_c	エンティティ c
(a)	Chikamatsu was born Sugimori Nobumori to a samurai family.	`Chikamatsu_Monzaemon`
(b)	There is disagreement about his birthplace.	`Chikamatsu_Monzaemon`
(c)	The most popular theory suggests he was born in Echizen Province, but there are other plausible locations, including Hagi, Nagato Province.	`Chikamatsu_Monzaemon`
(d)	Harry Potter is a series of fantasy novels written by British author J. K. Rowling.	`Harry_Potter`
(e)	The novels chronicle the life of a young wizard, Harry Potter, and his friends Hermione Granger and Ron Weasley, all of whom are students at Hogwarts School of Witchcraft and Wizardry.	`Harry_Potter`
(f)	The main story arc concerns Harry's struggle against Lord Voldemort, a dark wizard who intends to become immortal, overthrow the wizard governing body known as the Ministry of Magic, and subjugate all wizards and Muggles, a reference term that means non magical people.	`Harry_Potter`

拡張データのテキスト-エンティティ対のデータの例を表 3 に示す．表 3 の例では，(a) と (c) のテキストは，エンティティの `Chikamatsu_Monzaemon`（近松門左衛門）に関する情報が書かれている．一方，(b) のテキストは，それ自身には近松に関する情報は書かれていない．このように，本研究で作成した拡張データには，擬似的なクイズ問題とみなすにはノイズとなるような事例も含まれている．ノイズとなり得る文をフィルタリングで取り除いて拡張データの品質を上げることは，本研究の今後の課題である．

以上の手法により作成した拡張データを，本研究では TriviaBERT モデルの事前訓練に用いる．

5. 実験

本節では，QANTA データセット上での提案手法の有効性を検証する実験と，その結果について述べる．

5.1 共通設定

すべての実験では，2.1 節で述べた QANTA 2018 データセットを使用した．表 1 に示したとおり，QANTA の train データには，25,969 種類のエンティティが正解として出現する．本研究では，この train データに出現するエンティティの集合を，正解エンティティの集合 C とする（すなわち，25,969 クラス分類問題となる）．

TriviaBERT に含まれる BERT には，訓練済みのモデルとして公開されている $\text{BERT}_{\text{BASE}}$ を用いた（ベクトルの次元数 $d = 768$，層数 $L = 12$）．TriviaBERT が計算する文ベクトル $\mathbf{h}_x$ および各クラス（エンティティ）の重みベクトル $\mathbf{w}_c$ の次元数 d' は 256 とした．訓練時の負例エンティティのサンプル数 K は 100 とした．訓練時のミニバッチサイズは 256 とし，最適化は [3] と同様に行った．すなわち，アルゴリズムに Adam[10] を，勾配の clipping (1.0) と学習率の warmup を適用した．学習のステップ数と warmup のスケジュールはそれぞれの実験において異なるが，具体的な値は付録 A.1, A.2 に記載した．クイズ解答性能の評価指標には，Hit@1，Hit@10，Hit@100，および MRR（平均逆順位）を用いた．

表 4　test データにおける TriviaBERT の解答性能

	Hit@1	Hit@10	Hit@100	MRR
$l = 1$	**0.5383**	0.6776	0.7342	**0.5892**
$l = 6$	0.5081	0.7085	0.7816	0.5802
$l = 12$	0.4769	**0.7385**	**0.7948**	0.5406

5.2 モデルの有効性の検証

提案モデル TriviaBERT のクイズ解答性能を検証する．特に，TriviaBERT で問題ベクトル $\mathbf{h}_x$ の計算時（3.2 節 (4) 式）に使う BERT の出力層の数 l を 1，6，12 とした各場合について実験を行う．

実験結果を表 4 に示す．Hit@1 および MRR の指標では，BERT の最終層のみを使った（$l = 1$）TriviaBERT の性能が最高値を記録した．一方，Hit@10 および Hit@100 では，$l = 6$ および $l = 12$ の TriviaBERT の性能が，$l = 1$ の性能を上回った．各モデルの出力傾向を見てみると，層を多く使った TriviaBERT は，予測スコアの高い順にエンティティをソートした際，正解エンティティとそれに類似したエンティティが上位を占める傾向にあった． それらの類似したエンティティに最も高い予測スコアをつけ，正解エンティティとして選出してしまうケースが見られた．したがって，類似したエンティティ間の識別力を向上させることが今後の課題の一つである．一方で，正解エンティティに高い予測スコアをつけることには成功しているため，予測スコア上位 N ベスト解に基づくリランキング手法と組み合わせることで，より性能の良い解答システムを構築できる可能性も示唆している．

表 5　訓練データ拡張を行った TriviaBERT の解答性能

	Hit@1	Hit@10	Hit@100	MRR
Quiz	0.5081	0.7085	0.7816	0.5802
Wiki-lead	0.0068	0.0414	0.1501	0.0194
Wiki-all	0.1291	0.3667	0.6001	0.2087
Wiki-lead → Quiz	0.5412	0.7000	0.7766	0.5999
Wiki-all → Quiz	**0.6304**	**0.7717**	**0.8099**	**0.6845**

5.3　データ拡張の有効性の検証

Wikipedia 記事を利用したデータ拡張の有効性を検証するため，以下の 5 つの設定で実験を行った．

- **Quiz:** QANTA の train データのみでモデルを訓練する（5.2 節と同じ設定）．
- **Wiki-lead:** Wikipedia 記事の最初の段落のみから作成した擬似データのみでモデルを訓練する．
- **Wiki-all:** Wikipedia 記事の本文全文から作成した擬似データのみでモデルを訓練する．
- **Wiki-lead → Quiz:** Wikipedia 記事の最初の段落のみから作られた擬似データのみでモデルを訓練し，さらに QANTA の train データでモデルの fine-tuning を行う．
- **Wiki-all → Quiz:** Wikipedia 記事の本文全文から作成した擬似データでモデルを訓練し，さらに QANTA の train データでモデルの fine-tuning を行う．

一般に，Wikipedia 記事の最初の段落には，記事の最も重要な内容が書かれるため，最初の段落を構成する文のみから，クイズ解答に必要十分な情報を得られる可能性がある．Wiki-lead および Wiki-lead → Quiz は，この可能性を検証するための設定である．一方で，Quizbowl では，正解の事物についてのあらゆる側面の知識が問われるため，記事本文のすべての文の情報がクイズ解答の性能向上に寄与しうる．そのため，Wiki-all およびおよび Wiki-all → Quiz では，Wikipedia 記事中のすべての段落の文を用いている．

実験の結果を表 5 に示す．まず，データ拡張手法により自動生成した擬似データのみで訓練したモデルである Wiki-lead と Wiki-all を比較する．すべての評価指標で Wiki-all が高い性能を記録した．この結果は，Wikipedia 記事の最初の段落のみをデータ拡張に利用するよりも，すべての段落を利用した方が効果的であることを示している．次に，これらの擬似データのみで訓練したモデルと，Quizbowl の訓練データのみで訓練したモデル（Quiz）を比較すると，Wiki-lead・Wiki-all モデルは Quiz モデルより低い性能を示した．これは，本研究の擬似データ生成の手法では，クイズの解答に役立たないようなデータも含まれてしまうため，クイズ用に整備されたデータで訓練したモデルの性能には届かない結果となったと解釈できる．しかし，Quiz モデルには及ばないまでも，Wiki-lead・Wiki-all モデルがある程度のエンティティに関する知識を学習できていることが各指標からわかる．

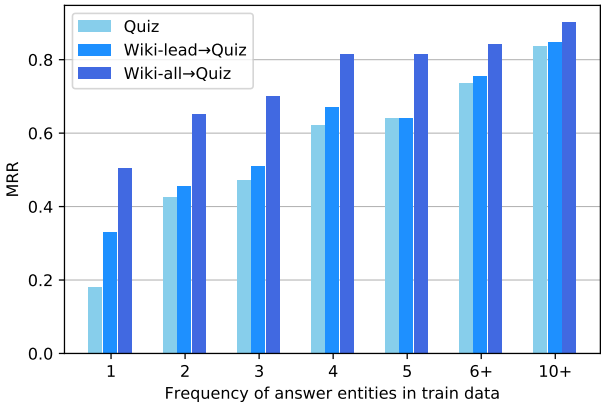

図 4　正解エンティティの train データにおける出現頻度と解答性能 (MRR) の関係

Wiki-lead と Wiki-lead → Quiz，Wiki-all と Wiki-all → Quiz を比較すると，どちらも fine-tuning したモデルの性能が大きく上回る結果となり，fine-tuning の有効性が示唆された．さらに，Quizbowl の訓練データのみで訓練したモデル（Quiz）と Wiki-lead → Quiz・Wiki-all → Quiz モデルを比較すると，Wiki-lead → Quiz・Wiki-all → Quiz モデルの性能が上回った．特に Wiki-all → Quiz モデルは Quiz モデルと比べ，Hit@1 と MRR で 10 ポイント以上の性能向上が見られた．この結果は，データ拡張手法によって生成された擬似データを用いてモデルの事前学習を行うことによって，より良いモデルパラメータの初期値が得られ，さらにそれをクリーンな訓練データで fine-tuning することによって，最終的により良い解答を出力するモデルを構築可能であることを示している．

6.　分析

TriviaBERT の性質をより明らかにするために，few-shot のテスト事例に対するモデルの性能と，問題の難易度ごとのモデルの性能を分析した．

6.1　few-shot のテスト事例に対するモデルの性能

モデルの解答性能を，正解エンティティの train データにおける出現頻度ごとに評価した結果を図 4 に示す．図 4 の実験結果は，拡張データでモデルの事前訓練を行う提案手法が，正解の訓練データにおける出現頻度が少ない問題，すなわち few-shot のテスト事例に対して，特に有効であることを示している．また，正解の訓練データにおける出現頻度が比較的多い問題に対しても，解答性能が向上しており，提案手法は，few-shot ではないテスト事例においても，一貫して解答性能の向上に寄与していることがわかる．

6.2　問題の難易度ごとのモデルの性能

2.2 節で述べたように，Quizbowl の問題は複数の文で構成されるが，1 つの文の情報だけでも正解を導けるように

表 6　文分割した test データにおける TriviaBERT の解答性能

	Hit@1	Hit@10	Hit@100	MRR
$l = 1$	0.0914	0.3534	0.6427	0.1746
$l = 6$	**0.1061**	0.3714	0.6497	0.1902
$l = 12$	0.1055	**0.3717**	**0.6514**	**0.1905**

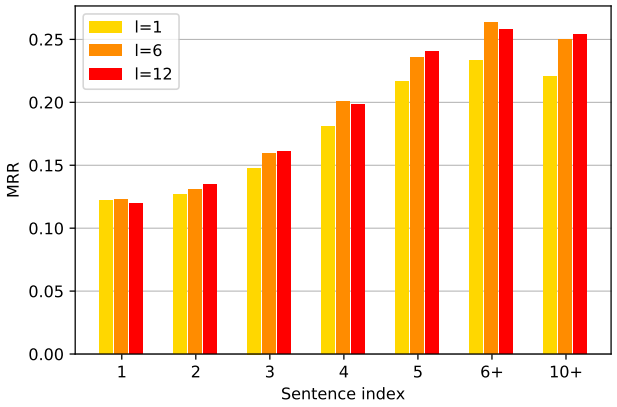

図 5　問題文の位置と解答性能 (MRR) の関係．問題文の位置を表す sentence index が大きいほど，問題文の難易度は低くなる．

作問されている．ただし，解答のための手がかりとなる情報が少なくなるため，問題文 1 文のみで解答するのは，問題全文を読んで解答するよりも難しい条件設定となる．このようなより難しい条件での提案モデルの性能を評価するため，train データおよび test データの各問題を文分割し，問題 1 文を 1 事例として訓練・評価する実験設定でモデルの分析を行う．

実験結果を表 6 に示す．文分割をして 1 文 1 事例とした場合には，$l = 6$ および $l = 12$ の TriviaBERT の性能が $l = 1$ とした場合を上回り，5.2 節の実験結果とは異なる傾向を示した．この傾向をより詳細に分析するため，問題文の位置（何文目の問題文であるか）ごとのモデルの解答性能を図 5 に示す．この実験結果は，問題文の位置が後ろになる（問題文の難易度が低くなる）ほど，モデルの解答性能が高くなることを示している．これは，Quizbowl の問題が持つ pyramidality の性質（2 節）が表れた結果となっている．

これらの結果を解釈するために，以下の仮説を立てることができる．問題を文分割して 1 文 1 事例とすると，難易度が高い問題文も低い問題文もそれ単体で訓練事例となるため，モデルにはどの難易度の問題文の入力に対しても正しく解答できることが要求される．そのような条件下では，BERT の最終層が出力する情報だけを用いて任意の難易度の問題文に解答できるようにモデルを訓練するよりも，複数の層が出力する情報を用いて解答できるように訓練する方が，可変の難易度の問題文の入力に対して解答性能が高くなると考えられる．この仮説は，ある定数 k について，モデルを train データの k 文目の問題文だけで訓練し，test データの k 文目だけで評価すれば，検証をすることができると考えられる．より詳細な分析は今後の課題である．

7. 関連研究

言語処理技術を用いてクイズ問題を解く研究は，その問題の捉え方により，多クラス分類問題としてのアプローチと，読解問題としてのアプローチに大別される．

7.1 多クラス分類問題としてのクイズ解答タスク

クイズ解答タスクを多クラス分類タスクとして解くアプローチは Boyd-Graber ら [7] によって提案された．彼らが提案した QANTA 2012 データセット，および Iyyer ら [11] が提案した QANTA 2014 データセットは，我々と同様に Quizbowl を題材としているが，コーパス内において一定回数以上出現する正解のみを解答候補とした，few-shot の状況が緩和されたデータセットとなっている．これらのデータセットを題材にした，国際的なコンペティションも開催されている． Yamada ら [12] は，正解のエンティティ分類および固有表現クラスの分類を行うニューラルモデルと，情報検索のモデルを組み合わせた手法により，早押しクイズの設定において人間のエキスパートを凌駕する性能を持つシステムを開発した．

Rodriguez ら [6] が提案した QANTA 2018 データセットは，Quizbowl の問題を用いたデータセットでは本稿執筆時点で最新のものである．従来のバージョンの QANTA データセットとは異なり，正解の出現回数の制約が取り払われた，few-shot の事例が存在するデータセットとなっている．また彼らは，QANTA 2018 データセット上で，情報検索の手法や双方向 GRU などのニューラルネットを用いた実験を行い，ベースラインとなる性能を報告している．さらに，Wikipedia の記事の最初の段落を拡張データとして訓練データに追加することも提案している．本研究におけるデータ拡張手法はこれを発展させ，対象を Wikipedia 記事全文に拡大し，モデルの事前訓練のために利用したものである．

このアプローチにおいては，解答は Wikipedia の記事になるような固有名詞に限られるものの，後述する読解問題としてのタスク設定に比べてシステム全体がシンプルなものとなる傾向がある．

本研究は，このアプローチにおいて汎用言語モデルとデータ拡張がどのようにクイズ解答性能の向上に寄与するのかを調べた初めての研究である．

7.2 読解問題としてのクイズ解答タスク

クイズ解答タスクのもう一つの主要なアプローチは，Wikipedia などのコーパスを読解対象文書集合とした読解タスク（スパン抽出タスク）として解く方法である．たとえば， TriviaQA [13] は，Web から収集したクイズの質問

文とその正解のペアに対して，質問に関連する Web ページと Wikipedia 記事を自動で付与することで作られた，およそ 65000 件の質問からなる読解データセットである．

このアプローチには，読解対象文書を適切に選択することができれば，固有名詞であるか否かを問わずに解答することができるという利点がある．しかしながら，このアプローチにおいては，読解対象の文書が解答に必要な情報を十分に含んでいるかどうか（別の言い方をすれば，読解対象文書に答えが含まれていないという状況をどのように扱うか）が問題になる．この問題に適切に対処するために，たとえば鈴木らは 解答可能性付き読解データセット を作成した [14]．これは 12000 件の日本語の早押しクイズの問題と正解に対して，関連する Wikipedia の記事段落を機械的に付与し，それぞれに対して読解での解答可能性を付与したデータセットである．また，機械読解のメジャーなデータセットの一つである SQuAD も，最近リリースされたバージョン 2.0 において，解答不可能であるという状況を正しく認識することが求められるようになった [15]．この問題設定における具体的なシステムにおいては，DrQA[16],ORQA[17],BERTserini[18] にみられるように，情報検索システムによる読解対象文書の選択と読解のパイプラインとして解く手法が主流である．

我々が今回取り扱った QANTA データセットにおいては，解答がすべて Wikipedia の記事と対応づく固有名詞であるということがわかっているため，今回はこのアプローチを採用しなかった．しかしながら，日本語のクイズ問題データセット [14] においては，固有名詞ではない正解も一定数存在するため，解答可能な問題のカバレッジを上げるために今後このアプローチを検討することも必要かもしれない．

7.3 汎用言語モデルのクイズ解答タスクへの応用

様々なタスクにおいて，事前訓練された言語モデルが性能向上に寄与することが報告されている．なかでも，代表的なモデルである BERT[3] は，読解問題をはじめとした様々なタスクにおいて有効性が検証されている．読解に基づく質問応答タスクにおいても，BERTserini [19] において，その有効性が示唆されているが，超多クラス分類問題としてのクイズ解答タスクへの応用は本研究が初めてのものである．

8. おわりに

本研究では， Quizbowl を題材に，クイズ解答タスクを超多クラスの分類問題とみなして解く方法について述べた．具体的には，汎用言語モデル BERT をもとに，超多クラス分類問題に対応したモデルである TriviaBERT を提案し，また，Wikipedia の記事を利用したデータ拡張方法を提案した．実験の結果，モデル・データの拡張の両方がクイズ解答性能の向上に寄与していることを明らかにした．

今後の課題として，以下の方向が挙げられる．

- **zero-shot 学習への応用．** 本稿では，Quizbowl の訓練データに出現する正解のみを解答候補としたが，この設定では，訓練データに一度も出現しない正解エンティティ，すなわち zero-shot の事例には答えることができない．そこで，データ拡張をさらに多くの Wikipedia の記事（究極的には全ての記事）に適用することで，zero-shot 学習が可能になると考えられる．
- **読解系モデルとの統合．** 5.2 節の結果から，BERT の複数の層の出力を用いた TriviaBERT は Hit@10 や Hit@100 の指標での解答性能が高く，リランキングを伴う QA のモデルとの親和性は高いと考えられる．あるいは，TriviaBERT を文書検索のモジュールとして用い，予測されたエンティティの文書を読解モデルに読ませることで，検索と読解を統合した型の QA のモデルに応用することも可能だと考えられる．

謝辞 本研究は JSPS 科研費 JP19H04162，JP19K20351 の助成を受けたものです．

参考文献

[1] Yampolskiy, R. V.: Turing Test as a Defining Feature of AI-Completeness, *Artificial Intelligence, Evolutionary Computing and Metaheuristics: In the Footsteps of Alan Turing* (Yang, X.-S., ed.), Studies in Computational Intelligence, Springer, Berlin, Heidelberg, pp. 3–17 (online), available from ⟨https://doi.org/10.1007/978-3-642-29694-9₁⟩ (2013).

[2] Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L.: Deep Contextualized Word Representations, *NAACL*, pp. 2227–2237 (online), DOI: 10.18653/v1/N18-1202 (2018).

[3] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *NAACL*, Vol. 1, pp. 4171–4186 (online), available from ⟨https://www.aclweb.org/anthology/N19-1423⟩ (2019).

[4] Sennrich, R., Haddow, B. and Birch, A.: Improving Neural Machine Translation Models with Monolingual Data, *ACL*, Vol. 1, pp. 86–96 (online), DOI: 10.18653/v1/P16-1009 (2016).

[5] Edunov, S., Ott, M., Auli, M. and Grangier, D.: Understanding Back-Translation at Scale, *EMNLP*, pp. 489–500 (online), DOI: 10.18653/v1/D18-1045 (2018).

[6] Rodriguez, P., Feng, S., Iyyer, M., He, H. and Boyd-Graber, J.: Quizbowl: The Case for Incremental Question Answering, *CoRR*, Vol. arXiv:1904.04792 (online), available from ⟨http://arxiv.org/abs/1904.04792⟩ (2019).

[7] Boyd-Graber, J., Satinoff, B., He, H. and Daum III, H.: Besting the Quiz Master: Crowdsourcing Incremental Classification Games, *EMNLP*, pp. 1290–1301 (online), available from ⟨https://www.aclweb.org/anthology/D12-1118⟩ (2012).

[8] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, *NIPS* (Burges,

C. J. C., Bottou, L., Welling, M., Ghahramani, Z. and Weinberger, K. Q., eds.), pp. 3111–3119 (online), available from ⟨http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf⟩ (2013).
[9] Jawahar, G., Sagot, B. and Seddah, D.: What Does BERT Learn about the Structure of Language?, *ACL*, pp. 3651–3657 (online), available from ⟨https://www.aclweb.org/anthology/P19-1356⟩ (2019).
[10] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *ICLR*, (online), available from ⟨http://arxiv.org/abs/1412.6980⟩ (2015).
[11] Iyyer, M., Boyd-Graber, J., Claudino, L., Socher, R. and Daum III, H.: A Neural Network for Factoid Question Answering over Paragraphs, *EMNLP*, pp. 633–644 (online), DOI: 10.3115/v1/D14-1070 (2014).
[12] Yamada, I., Tamaki, R., Shindo, H. and Takefuji, Y.: Studio Ousia's Quiz Bowl Question Answering System, *CoRR*, Vol. arXiv:1803.08652 (online), available from ⟨http://arxiv.org/abs/1803.08652⟩ (2018).
[13] Joshi, M., Choi, E., Weld, D. and Zettlemoyer, L.: TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension, *ACL*, Vol. 1, pp. 1601–1611 (online), DOI: 10.18653/v1/P17-1147 (2017).
[14] 鈴木正敏，松田耕史，岡崎直観，乾健太郎：読解による解答可能性を付与した質問応答データセットの構築 (2018).
[15] Rajpurkar, P., Jia, R. and Liang, P.: Know What You Don't Know: Unanswerable Questions for SQuAD, *ACL*, Vol. 1, Melbourne, Australia, Association for Computational Linguistics, pp. 784–789 (online), available from ⟨https://www.aclweb.org/anthology/P18-2124⟩ (2018).
[16] Chen, D., Fisch, A., Weston, J. and Bordes, A.: Reading Wikipedia to Answer Open-Domain Questions, *ACL*, Vol. 1, pp. 1870–1879 (online), DOI: 10.18653/v1/P17-1171 (2017).
[17] Lee, K., Chang, M.-W. and Toutanova, K.: Latent Retrieval for Weakly Supervised Open Domain Question Answering, *ACL*, pp. 6086–6096 (online), available from ⟨https://www.aclweb.org/anthology/P19-1612⟩ (2019).
[18] Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M. and Lin, J.: End-to-End Open-Domain Question Answering with BERTserini, *NAACL*, Vol. Demonstrations, pp. 72–77 (online), DOI: 10.18653/v1/N19-4013 (2019).
[19] Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M. and Lin, J.: End-to-End Open-Domain Question Answering with BERTserini, *NAACL-HLT* (2019).

付　　録

A.1　モデルの有効性検証実験の設定詳細

それぞれの設定で train データで TriviaBERT のモデルを訓練し，訓練したモデルを test データに適用してモデルの解答性能を評価した．訓練のステップ数は 50,000 とし，最初の 5,000 ステップで学習率の warmup を行った．

A.2　データ拡張の有効性検証実験の設定詳細

それぞれの設定で train データで TriviaBERT のモデルを訓練し，訓練したモデルを Quizbowl の test データに適用した．TriviaBERT のモデルは，5.2 節の実験で性能のバランスの良かった $l = 6$ のモデルを用いた．Wikipedia データセットの訓練のステップ数は 100,000 とし，最初の 10,000 ステップで学習率の warmup を行った．Quizbowl データセットの訓練のステップ数は 50,000 とし，最初の 5,000 ステップで学習率の warmup を行った．