JAQKET: クイズを題材にした日本語 QA データセットの構築

鈴木 正敏 [1,2*] 鈴木 潤 [1,2*] 松田 耕史 [2,1] 西田 京介 [3] 井之上 直也 [1,2]
[1] 東北大学 [2] 理化学研究所 [3]NTT メディアインテリジェンス研究所
{m.suzuki,jun.suzuki,matsuda,naoya-i}@ecei.tohoku.ac.jp, kyosuke.nishida.rx@hco.ntt.co.jp

1 はじめに

計算機により自然言語を正しく理解する方法論を確立することは，自然言語処理および人工知能研究の最終目的の一つである．その実現に向けた取り組みとして，質問応答や機械読解に関する研究が世界的にみて現在大きな注目を集めている．その証拠として，例えば，SQuAD [6], TriviaQA [3], SearchQA [2], QANTA [7] など，多くの質問応答/機械読解の評価用データセットがここ数年で新たに構築されており，様々な観点から，現在の技術で「どこまで計算機が言語を扱えるか」を評価する試みがなされている．特に，これらのデータセットは，2018 年に ELMo[5] の発表を発端に急速に発展した，大規模文章集合から学習したニューラル言語モデルを用いた方法論が，言語知識をどの程度うまく保持しているかを計測する評価用データとしても利用されるようになり，ニューラル言語モデルの発展とともに，質問応答/機械読解の研究も日進月歩で発展している．

ただし，こういった最新の質問応答/機械読解用のデータセットは，英語を対象言語としたものがほとんどであり，日本語による同様のデータセットで現在広く用いられているものは存在しない．そこで，本研究では，日本における質問応答/機械読解研究の促進を目的として，研究者が容易に利用可能な日本語のオープンドメイン QA タスクのデータセット**「JAQKET」**[*1]を構築する．作成するデータセットは，既存研究 [7] に倣い，Wikipedia[*2]の記事名を答えとした，日本語のオープンドメイン QA タスクのデータセットである．[*3]

2 関連研究

近年の QA 研究の拡大の背景として，大規模な QA データセットが多数登場したことが挙げられる．

2016 年に公開された SQuAD [6] は，Wikipedia の記事に対して人手で作られた 10 万問以上の質問からなる，読解型 QA の大規模なデータセットである．SQuAD では，データセットだけでなく，提出された QA システムの解答性能を順位付けして示すリーダーボードも提供されており，より性能の高い QA システムの研究を促進する環境が用意されていた．SQuAD の公開以降，世界中の研究者が SQuAD の読解型 QA タスクに取り組み，その結果，BiDAF [8] や QANet [10] に代表される多くの読解型 QA の手法が提案された．

SQuAD の登場以降，さらに多様なドメイン・形式の QA タスクおよび大規模なデータセットが数多く提案されるようになった．その中には，クイズ問題を利用して作成されたデータセットがいくつか存在する．TriviaQA [3] は，クイズ問題集の Web サイトから収集されたクイズ問題に Web ページや Wikipedia 記事を文書として付与した，読解型 QA のデータセットである．SearchQA [2] は，テレビ番組 Jeoperdy! で出題されたクイズ問題に対して，Google の検索結果のスニペットを文書として付与した，読解型 QA のデータセットである．QANTA [7] は，Quizbowl という英語圏で行われている早押しクイズの問題を用いて作られたオープンドメイン QA のデータセットで，問題の正解が全て Wikipedia の記事タイトルに単一化されている．クイズ問題以外にも，RACE [4] に代表される英語の試験問題を用いて作られた QA データセットがある．

以上のような，機械学習による手法を前提とした大規模な QA データセットは，そのほとんどが英語を対象言語としたものであり，日本語による同様の QA データセットは少ない．鈴木ら [11] は，日本語のクイズ問題およそ 1 万 2 千問に対して Wikipedia 記事を文書として付与し，文書の読解による解答可能性をクラウドソーシングにより付与した，解答可能性付き読解データセットを作成した．

3 タスク定義

本研究で扱う日本語オープンドメイン QA タスクを定義する．本研究では，クイズの問題文に対して複数（数個から数十個程度）の解答の選択肢が与られ，その選択肢から正解を一つ選択するという**択一問題**を取り扱う．以下にタスクおよびデータの構成要素を示す．

1. Q: 問題文（または，クイズの問題文）
2. A: 正解（または，問題文に対する正解）

*Equal contribution
[*1] JApanese Questions on Knowledge of EnTities
[*2] 日本語版 Wikipedia の 2019 年 1 月 21 日付の Cirrussearch ダンプファイルを用いた．
[*3] 本稿で説明するデータセット **JAQKET** の獲得方法および詳細な情報は以下のサイトに掲載
https://www.nlp.ecei.tohoku.ac.jp/projects/jaqket/

3. C: 選択肢（または，解答候補）
4. P: 記事（または，選択肢に関する記事）

次に，一つのクイズ問題 $\mathcal{Z}$ は，それぞれ一つの Q，A と K 個の C，P で構成されると定義する．つまり，$\mathcal{Z} = (Q, A, \{C_k, P_k\}_{k=1}^{K})$ である．

ここで，同一クイズ問題 $\mathcal{Z}$ 中の Q と A，または，P_k と C_k はそれぞれ対応関係にあると考える．また，選択肢 C の集合には正解 A が必ず含まれていると仮定する．つまり，$A \in \{C_k\}_{k=1}^{K}$ である．よって，学習用データ $\mathcal{D}^{\text{trn}}$ は，$\mathcal{D}^{\text{trn}} = \left\{\left(Q_i^{\text{trn}}, A_i^{\text{trn}}, \{C_{i,k}^{\text{trn}}, P_{i,k}^{\text{trn}}\}_{k=1}^{K}\right)\right\}_{i=1}^{N}$ と定義できる．このとき，開発用データ $\mathcal{D}^{\text{dev}}$ および評価用データ $\mathcal{D}^{\text{evl}}$ も $\mathcal{D}^{\text{trn}}$ と同様の形式となる．

解答に関する制限．本研究では，正解 A の構成要素を Wikipedia 記事名に限定する．こうすることで，モデルの出力を評価する際に，表記揺れを考慮する必要がなくなり，正解率による明解な評価が可能という利点もある．また，モデルを訓練する際に，Wikipedia 記事の本文，画像，カテゴリ情報などのメタデータを直接利用する発展も考えられる．

解答候補に関する制限．正解 A と同様に，選択肢 C も Wikipedia 記事名に限定する．よって，選択肢数 K の上限は Wikipedia の記事数である．ただし，日本語 Wikipedia にはおよそ 100 万記事が存在する．単純に $K = 1,000,000$ のような設定は，計算量的に破綻する可能性が高く現実的ではない．本研究では，人間が択一問題のクイズに答える際と同様の設定として，解答候補 K を数個から数十個程度に制限する．

4 データセットの構築

前節で述べたタスク定義に従って，タスク達成度を評価するための開発/評価用データ，および，自動解答モデルの訓練に必要となる学習用データを構築する．本節では，これら実際に作成した開発/評価用，および，学習用データに関して，作成手順と作成した実例を述べる．

4.1 開発/評価用データ

開発/評価用データは，タスク達成率を正確に測定するために，クイズ問題を作成することを専門とするクイズ作家にクイズ問題の作成を依頼し作成した．ただし，具体的には，問題文 Q と対応する正解 A の部分のみの作成を依頼した．

具体的な作業手順として，まず事前に送付した Wikipedia 記事名の集合から任意の一つを選択し，その記事名が正解 A となる問題文 Q を作成する，という手順でクイズ問題の作成を実施する．このような手順をとることで，3 節で述べたタスク定義に違反しない Q と A のペアを作成することができる．

ただし，この段階では，正解 A の Wikipedia 記事の本文中に問題文 Q を解答するために必要な全ての手がかりが含まれていることは保証していない．

4.2 学習用データ

開発/評価用データとは別に，より大規模な学習用データも作成した．学習用データは，Web で公開されている既存のクイズ問題に，答えの Wikipedia 記事を自動的に付与することによって作成した．

クイズ問題集の Web サイト『クイズの杜』[*4]と『abc/EQIDEN 公式サイト』[*5]より，クイズ大会「abc/EQIDEN」で 2003 年から 2014 年までの間に使用された全てのクイズ問題を収集した．正誤表[*6]に基づき一部の問題の訂正・除去を行った結果，問題数は 17,735 問となった．

開発/評価用データの作成時には，初めから Wikipedia の記事名が正解 A となるように問題を作成していた．これに対し，学習データのクイズ問題に対しては，既に与えられている正解に対して，適合する Wikipedia 記事を自動で付与する．ここで問題となるのは，Web サイトから収集されたクイズ問題には，「手塚治虫（てづか・おさむ）」「高床式倉庫【「高床倉庫」も ○】」のように，正解が読み仮名や注釈が付記されて記述されていることである．したがって，正解の記述と Wikipedia 記事名との完全一致だけでは，正解として適切な記事名を付与できない場合がある．そこで，正解の記述からルールベースの処理によって注釈部分を抜き出し，Wikipedia 記事を検索することで，検索された（またはリダイレクトされた）記事のタイトルを，それぞれのクイズ問題の正解 A として自動的に付与した．

4.3 選択肢 C の付与

3節で述べたように，本研究では択一問題を対象とする．そのため，各クイズ問題 $\mathcal{Z}$ に対して，解答の選択肢 C の付与を行う．

具体的には，日本語 Wikipedia エンティティベクトル [9] のコサイン類似度を Wikipedia 記事間の類似度と考え，各問題の正解 A ついて，正解 A の記事と類似度が高い記事を上位 1,000 件抽出した．抽出された記事のうち，正解 A と固有表現分類が同じもの，または類似度が上位である 20 件を，その問題の選択肢 C として付与した．固有表現分類のデータには，拡張固有表現分類 ver.8[*7]を用いた．

4.4 データの統計

表 1に作成したデータ量を示す．開発/評価用データは独立に 2 社のクイズ作家にデータを作成してもらったため，セット A，B の二種類が存在する．学習用データ，開発/評価用データは，それぞれ別の方法で作成したにもかかわらず，おおよそ同じ平均文字数になった．このことから，質問文に含まれる文字列としての情報量は，それぞれのデータの傾向が大きく逸脱している

[*4] https://quiz-schedule.info/quiz_no_mori
[*5] http://abc-dive.com
[*6] http://abc-dive.com/questions/errata.html
[*7] http://shinra-project.info/download/

表1　問題文 Q に関する統計量

	文数	文字数	平均文字数/文
学習用データ	12,019	586,278	48.8
開発用データ A	996	48,759	49.0
評価用データ A	998	51,938	52.4
開発用データ B	991	51,342	51.4
評価用データ B	999	51,692	51.7

表2　質問タイプ毎の質問数および割合

			開発/評価用セット A				開発/評価用セット B			
タイプ	学習	(%)	開発	(%)	評価	(%)	開発	(%)	評価	(%)
なに	4,185	(34.8)	662	(66.5)	740	(74.1)	482	(48.6)	469	(46.9)
なに〜	3,611	(30.0)	80	(8.0)	74	(7.4)	309	(31.2)	299	(29.9)
だれ	2,468	(20.5)	151	(15.2)	117	(11.7)	124	(12.5)	136	(13.6)
どこ	1,391	(11.6)	88	(8.8)	59	(5.9)	59	(6.0)	79	(7.9)
どんな	189	(1.6)	7	(0.7)	3	(0.3)	16	(1.6)	11	(1.1)
いくつ	42	(0.3)	2	(0.2)	1	(0.1)	0	(0.0)	0	(0.0)
どの	129	(1.1)	6	(0.6)	4	(0.4)	1	(0.1)	5	(0.5)
いくら	2	(0.0)	0	(0.0)	0	(0.0)	0	(0.0)	0	(0.0)
いつ	2	(0.0)	0	(0.0)	0	(0.0)	0	(0.0)	0	(0.0)
合計	12,019		996		991		998		999	


```
{
  "qid": "QA20QBIK-0002",
  "question": "童謡『たなばたさま』の歌詞で、「さらさら」と歌われる植物は
   何の葉？",
  "answer_entity": "ササ",
  "answer_candidates": [ "ササ", "チシマザサ", "クマザサ", "アダン",
    "チガヤ", "アセビ", "ススキ", "ホオノキ", "マテバシイ", "ヤマフジ",
    "ウツギ", "タムシバ", "ミズキ", "アキタブキ", "トベラ", "クヌギ",
    "ネズミモチ", "ヒシ", "コブシ", "オオウバユリ" ],
  "qtype": "なに〜",
}
{
  "qid": "QA20QBIK-0026",
  "question": "北海道の中心に位置することから「北海道のへそ」を名乗る、
    ラベンダーで有名な都市はどこ？",
  "answer_entity": "富良野市",
  "answer_candidates": [ "富良野市", "滝川市", "北見市", "芦別市",
    "中富良野町", "名寄市", "網走市", "美瑛町", "南富良野町", "岩見沢市",
    "美唄市", "上富良野町", "倶知安町", "小樽市", "歌志内市", "旭川市",
    "ニセコ町", "北斗市", "稚内市", "帯広市" ],
  "qtype": "どこ",
}
```


図1　実データの例 (jsonl 形式)

ということがないということが確認できる．

次に，表 2に「何に対する質問か」を粗く分類した結果を示す．ここでは，「なに」「なに〜」[*8]「だれ」「どこ」「どんな」「いくつ」「どの」「いくら」「いつ」の 9 種類の質問タイプで分類した．分類には，簡単な表層マッチングによる一次分類と人手による最終チェックにより分類を行った[*9]．

最後に，図 1に最終的に作成したデータ（jsonl 形式）のサンプルを示す．`"qid"` は一意に与えられた ID, `"question"` は問題文，`"answer_entry"` は正解，`"answer_candidates"` は選択肢，`"qtype"` は補足情報の質問タイプの情報である．

5　実験

作成した学習/開発/評価用のデータに対して，現状の技術を用いてどの程度問題が解けるかを検証することで，当該データセットの難易度の検証を行った．

表3　学習時のハイパーパラメタ

項目	値	項目	値
埋め込み層次元数	768	隠れ層数	12
隠れ層次元数	768	dropout 率	0.1
FF 層次元数	3072	学習率	0.00005
注意機構 head 数	12	epoch 数	5

5.1　複数選択肢クイズ解答手順

本稿では，文献 [1] で提案された事前学習済み言語モデル BERT，および，BERT を用いて様々な自然言語処理タスクの性能を評価した手順を踏襲し実験を行った．

具体的には，事前学習済み言語モデルとして，`huggingface/transformers`[*10]に組み込まれている日本語 BERT モデルを採用した．学習の手順としては，文献 [1] の実験同様，事前学習済み言語モデルを初期値とし，4 節で説明した学習用データを用いてモデルを fine-tuning する形式でクイズ解答用のモデルを学習した．このとき，最終的なモデルは複数選択肢解答用のモデルを学習する．複数選択肢から，解答を選択する手順は以下のとおりである．

1. 選択肢 $\{C_k\}_{k=1}^K$ と対応する記事 $\{P_k\}_{k=1}^K$ を取得
2. 各 k に対し，(記事 P_k，問題文 Q，選択肢 C_k）の順番に文章を結合し，モデルの入力とする．
3. 各入力に対し，確率を取得
4. 最も高い確率となった選択肢 C_k を問題 Q の予測解答 $\widehat{Y}$ として出力

もし予測解答 $\widehat{Y}$ が正解 A と一致したらそのクイズ問題は正解と判定し，もし一致しなかったら不正解とする．

5.2　実験設定

選択肢の数 K は 20 とした．ただし，学習時は，GPU のメモリの都合上，選択肢 C の数 K を 5 に設定した．[*11] また，入力文を構成する最小単位を，文字単位とサブワード単位の 2 種類のモデルを用いて実験を行った．文字単位の場合の語彙数は 4,000，サブワード単位の場合の語彙数は 32,000 である．これは，実験で用いた学習済み日本語 BERT モデルの設定から自動的に決まるパラメタである．表 3に，モデル学習時に用いた人手調整のハイパーパラメタの一覧を示す．

5.3　結果

表 4に実験結果を示す．まず，全体の傾向として，大規模文章集合を用いて学習した学習済みニューラル言語モデルを初期値として，対象タスクのデータでファインチューニングするという近年の自然言語処理タスクで成功を収めている方法論を用いることで，人間が普

[*8] 「なに条約」「なに地方」といった「なに」の後に付加情報がついている質問を「なに〜」と分類した．付加情報がない場合は質問タイプ「なに」に分類した．

[*9] 質問タイプの分類も実験結果の詳細な分析に役立つと考え，それぞれのデータに情報として付与した．

[*10] `https://github.com/huggingface/transformers`

[*11] 本稿で用いたモデルでは，選択肢 C 全体や相対的な情報を用いていない方法論のため，学習と評価時で K が違っていても不都合は生じない．

表4　実験結果: 正解率

	開発/評価用セット A		開発/評価用セット B	
	開発	評価	開発	評価
文字単位	72.3	82.4	76.8	78.2
サブワード単位	76.6	83.0	82.3	82.5

表5　入力を (記事 P_k, 問題文 Q, 選択肢 C_k) ではなく，問題文 Q を用いずに（記事 P_k, 選択肢 C_k）でモデルを学習/評価した場合の正解率

	開発/評価用セット A		開発/評価用セット B	
	開発	評価	開発	評価
文字単位	25.9	26.7	32.4	30.3
サブワード単位	23.8	23.4	26.3	27.7

通に取り組むことができるクイズ問題を 8 割程度正解することができるという結果になった．次に，今回実験に用いた文字単位かサブワード単位かという比較では，サブワード単位のほうが明確に良い結果が得られた．

5.4　分析

データの品質検証． 表 5に，入力を (記事 P_k, 問題文 Q, 選択肢 C_k) ではなく，問題文 Q を用いずに（記事 P_k, 選択肢 C_k）で構築した場合の実験結果を示す．問題文 Q を使わない設定とは，すなわち，問題文 Q と記事 P_k 間の照合を行わなくても問題が解けるということを意味する．つまり，本質的に別の何かの手がかりが，本データセットのクイズ問題を解くことに寄与していることを意味する．

問題文 Q を用いないことで正解率が大幅に低下したという結果から，本データセットは，きちんと問題文 Q と記事 P_k 間の照合を行わないと，問題が解けないという本来の趣旨を満たしたデータセットとなっていることを確認できた．しかし，一方で，選択肢の数 K が 20 であり，チャンスレートが $1/20 = 5\%$ であることを考えると，20〜30% の正解率はそれなりに高いという解釈もできる．ただしこれは，クイズ問題は「世界一の××」や「×× で有名な」といった，クイズ問題の解答として選ばれやすい記事名は確実にあると言えるため，こういった偏りがある程度の手がかりを与えていると考えられる．今後は，こうした不要なバイアスを除去し，よりよいデータセットへの改良も検討したい．

学習時間． 国立研究開発法人産業技術総合研究所が構築・運用する AI 橋渡しクラウド ABCI[*12]，および，Google Colaboratory[*13]の二つの環境で実験を行った．[*14] ABCI 上で，4GPU(1 ノード) を用いた場合，学習用データによるモデルのファインチューニングにおよそ 50〜60 分程度かかった．同様の実験設定で Google Colaboratory 上で，1GPU を用いた場合は，およそ 5〜6 時間程度かかった．計算リソースが少ない研究者でも，本データを用いて研究をおこなうことが比較的容易であることが示せた．

[*12] https://abci.ai/ja/

[*13] https://colab.research.google.com/

[*14] ABCI および Google Colaboratory 上で利用できる GPU は，それぞれ nVIDIA Tesla V100 16GB メモリ，および，K80 8GB メモリである．

6　おわりに

本稿では，クイズを題材にした日本語 QA データセット「JAQKET」に関して，その構築法と構築したデータの統計量，分析結果の報告を行った．今後は，このデータを使って日本語における質問応答/機械読解の方法論の研究開発のベンチマークデータとして活用していきたいと考えている．また，このデータを使って自動クイズ解答システムに関するコンペティションの企画なども考えていきたい．

謝辞　本研究の一部は JSPS 科研費 JP19H04162, JP19J13238 の助成を受けたものです．本研究で使用した学習用クイズ問題は，abc/EQIDEN 実行委員会より研究目的での利用許可を頂きました．また，開発/評価用クイズ問題は，株式会社キュービックおよびクイズ法人カプリティオへ依頼して作成しました．記して感謝いたします．

参考文献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, volume 1, pages 4171–4186, 2019.

[2] Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. *CoRR*, arXiv:1704.05179, 2017.

[3] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *ACL*, volume 1, pages 1601–1611, 2017.

[4] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *EMNLP*, pages 785–794, 2017.

[5] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. In *NAACL*, pages 2227–2237, 2018.

[6] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *EMNLP*, pages 2383–2392, 2016.

[7] Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan Boyd-Graber. Quizbowl: The Case for Incremental Question Answering. *CoRR*, arXiv:1904.04792, 2019.

[8] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional Attention Flow for Machine Comprehension. In *ICLR*, 2017.

[9] Masatoshi Suzuki, Koji Matsuda, Satoshi Sekine, Naoaki Okazaki, and Kentaro Inui. A Joint Neural Model for Fine-Grained Named Entity Classification of Wikipedia Articles. *IEICE Transactions on Information and Systems*, E101.D(1):73–81, 2018.

[10] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In *ICLR*, 2018.

[11] 鈴木正敏, 松田耕史, 岡崎直観, and 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. In 言語処理学会第 *24* 回年次大会, pages 702–705, 2018.