

競技クイズ・パラレル問題の基本構造と文型

橋元 佐知 佐藤 理史 宮田 玲 小川 浩平

名古屋大学大学院工学研究科

hashimoto.sachi@d.mbox.nagoya-u.ac.jp

1 はじめに

与えられた質問に対して答を導く質問応答 (QA) タスクでは、質問としてクイズの問題がしばしば用いられる。クイズ問題を解くシステムとしては、クイズ番組 Jeopardy! で人間のチャンピオンを破った IBM の Watson が有名である [1]。この他にも、TriviaQA [2], SearchQA [3], QANTA[4] など、クイズの問題を用いて作成された QA タスク評価用データセットが存在する。これらは英語のデータセットであるが、日本語でも、JAQKET というデータセットが作成されている [5]。

一般に、QA タスクでは、問題文がテキストとして完全に与えられるのが普通である。しかしながら、人間がクイズに解答する状況は、これとは異なる場合がある。問題文が読み上げられる形式の早押しクイズでは、解答者は、他の解答者に先んじるために、問題文が最後まで読み上げられる前に答えることがしばしばある。このような場合、解答者は、完結しない問題文から答を導くことが必要となる。

このような読み上げ・早押しクイズ特有のひっかけ問題に、以下のようなパラレル問題がある。

- (1) アイルランドの首都はダブリンですが、アイスランドの首都はどこでしょう？

たとえば、競技クイズの大会の一つである abc/EQIDEN¹⁾ の過去問では、全体の約 11% がこのようなパラレル問題である。このような問題に「アイルランドの首都は」の段階で答えると、「ダブリン」という誤答になりそうである。

しかしながら、優れた解答者は、その段階で正しい答を導くことが多い。実際、「12 時間ガチクイズ生放送」という YouTube 動画でパラレル問題の解答状況を調査したところ、95 問中 63 問が「アイルランドの首都で」に相当する段階で解答されており、そのうちの 54 問が正解であった。

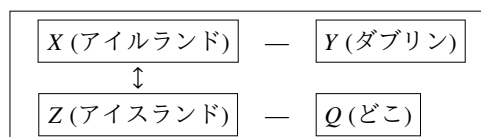


図1 パラレル問題の4項構造

ここで、次のような疑問が生じる。

1. なぜ、問題文の途中までで、高い精度で正しく答えることができるのか？
2. これを機械的に実現することは可能か？

これらの疑問に答えるために、我々は、まず、パラレル問題の基本構造と文型を調査した。そして、その結果に基づいて、問題文の途中までで答を導く解答システムを試作した。本論文では、これらの内容について報告する。

2 パラレル問題の基本構造と文型

パラレル問題の基本構造と文型を把握するために、abc/EQIDEN の過去問²⁾ に含まれる約 1,000 問のパラレル問題を調査した。その結果、以下の観察結果を得た。

1. パラレル問題の基本構造は、図1に示すような4項構造である。

本論文では、これらの4項を順に、 X, Y, Z, Q で表す。これらには、次の関係がある。

- i. 対比関係: $X \leftrightarrow Z$
 - ii. 同一関係: $X : Y = Z : Q$
2. パラレル問題を成立させているのは、 X と Z の対比関係である。

主要な対比関係の種類は、表1に示す5種類に整理できる。

3. 問題文は、図2に示すような4つのパートから構成される。

これらを順に、A, B, C, D パートと呼ぶ。A, C パー

1) <http://abc-dive.com>

2) <http://abc-dive.com/questions/>

表1 対比関係の種類

種類	具体例
1. 対義	大きい ↔ 小さい, 最初 ↔ 最後, 北 ↔ 南
2. 連続	一番目 ↔ 二番目, 春 ↔ 夏, 幼虫 ↔ さなぎ
3. 慣例的に一対一対応	衆議院 ↔ 参議院, 芥川賞 ↔ 直木賞, 五・一五事件 ↔ 二・二六事件
4. 共通の狭い集合	サッカー ↔ フットサル, キリスト教 ↔ イスラム教, 囲碁 ↔ 将棋
5. 音の類似・部分一致	アイルランド ↔ アイスランド, 戦争と平和 ↔ 戦争と平和の法, おおいぬ座 ↔ こいぬ座

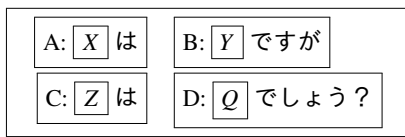


図2 問題文のパート構成

トの末尾は、典型的には提題表現(提題助詞「は」)であり、Bパートの末尾は、典型的には並列節の節末表現(接続助詞「が」)である。これらの強い手がかりにより、人間は、各パートの終了点を正しく認識できると考えられる。

以上の観察に基づき、問題文の途中までで正解を導くために必要なことを整理すると、以下のようになる。

1. Cパートまでで答える
すべての情報が揃っているので、通常の問題と同じように解ける
2. Bパートまでで答える
Xと対比関係にあるZを推測できれば、通常の問題と同じように解ける
3. Aパートまでで答える
Zに加え、XからYを推測し、その間の関係を同定できれば、通常の問題と同じように解ける

3 過去問のデータベース化

解答システムを作成する準備として、パラレル問題の過去問にアノテーションを施し、データベースを作成した。アノテーションでは、問題文を4つのパートに分割して末尾の正規化を行った後、表2に示す要素をマークアップした。アノテーションの具体例を図3に示す。

このようにマークアップしたテキストから、データベースのレコードを作成する。レコードの具体例

表2 マークアップする要素

タグ	内容
X	Aパートに含まれる項(対比要素)
Y	Bパートに含まれる項
Z	Cパートに含まれる項(対比要素)
Q	Dパートに含まれる項(疑問表現)
D	(問題の対象)ドメイン
x	Xの(広義の)上位概念
y	Yの(広義の)上位概念
z	Zの(広義の)上位概念
W	答
w	答Wの(広義の)上位概念

A: [X:アイルランド]の[y:首都]は
B: [Y:ダブリン]だ
C: [Z:アイスランド]の[y:首都]は
D: [Q:どこ]だ
答 [W:レイキャビク]

図3 問題文のアノテーション

表3 データベースのレコード形式

フィールド	(順方向)	(逆方向)
X	アイルランド	アイスランド
x		
Y	ダブリン	レイキャビク
y	首都	首都
D		
文型A	Xのyは	Xのyは
文型B	Yだ	Yだ
Z	アイスランド	アイルランド
z		
Q	どこ	どこ
W	レイキャビク	ダブリン
w	首都	首都
文型C	Zのyは	Zのyは
文型D	Qだ	Qだ
方向	0	1

を表3に示す。この例に示すように、レコードの作成では、以下のことを行う。

1. 順方向のレコードと逆方向の(A,Bパートと、C,Dパートを入れ替えた)レコードを作成する。
2. A,B,C,Dの各パートの文型パターンも登録する。
3. xとz、および、yとwは、一方が空である場合は、他方で補完する(下線部)。

4 解答システム

作成した解答システムの構成を図4に示す。本システムは、前節で示したデータベースを検索することで、問題の答を求める。システムは、以下のモジュールから構成されている。

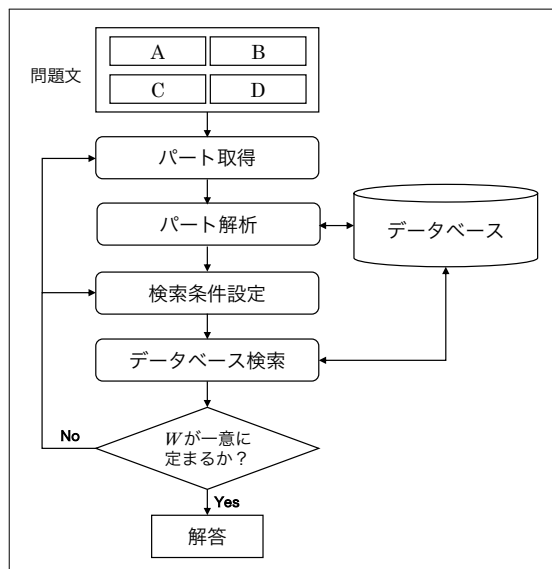


図4 解答システムの構成

1. パート取得

問題文の各パートを順に取得する。まず、Aパートのみを取得し、Bパート以降は、要求があった場合のみ取得する。

2. パート解析

取得したパートを解析し、表2のマークアップ要素と文型に分解する。この分解には、データベースに格納されている、該当パートの文型パターンを利用する。一般に、複数の文型パターンが存在するので、可能な分解をすべて求める。

3. 検索条件設定

得られた要素や文型、後述の解答判定により得られるフィードバック情報を用いて、データベースを検索する条件を設定する。

4. データベース検索

設定した検索条件で、データベースを検索し、得られたレコードから答Wを取得する。

5. 解答判定

Wの種類に応じて、以下のように分岐する。

(a) Wが一意に定まる場合

Wを解答として出力する

(b) Wが複数得られる場合

検索条件を強めるフィードバックを生成する

(c) Wがひとつも得られない場合

検索条件を緩めるフィードバックを生成する

検索条件を強めるとは、検索条件に新たな要素を追加することを意味する。これには、次のパートを取得して要素を追加することも含まれる。一方、検

表4 検索条件に含める要素

要素	取得済パート			
	A	B	C	D
X	r	r	(r)	(r)
Y		r	(r)	(r)
Z			r	r
D	o	o	o	o
x	o	o		
y	o	o		
z			o*	o*
w			o*	o*
文型A	a	a	a	a
文型B		a	a	a
文型C			a	a
文型D				a

索条件を緩めるとは、検索条件に含まれるいずれかの要素を削除することを意味する。表4に、それぞれのパートまで取得した場合に、どのような要素を検索条件に含めるかを示す。

この表の記号は、以下のことを意味する。

記号 **r** (required) 検索条件に必ず含める。

記号 **o** (optional) 取得済のパートに存在する場合に含める。検索条件を緩める場合は、この要素を削除する。削除した場合は、それ以降は使用しない。

記号 **a** (additional) 検索条件を強める場合に追加する。追加によってWがひとつも得られなくなる場合は、それ以降は使用しない。

たとえば、AパートからXとyが得られ、これらで検索すると複数のWが得られたとしよう。その場合は、まず、文型Aを検索条件に追加する。それでもWがひとつに絞れない場合に、Bパートを取得し、そこから検索条件に追加する要素を取得する。なお、文型Aを追加してWがひとつも得られなくなった場合は、文型Aを除外して、Bパートに進む。

一方、AパートからXとyとDが得られ、これらで検索するとWが得られない場合は、要素yを削除した条件、および、Dを削除した条件で検索し、それらの結果をマージして次の判断を行う。なお、XだけでもWがひとつも得られない場合は、Cパートまで取得して、それ以降の検索では、XとYを条件に含めない。「(r)」は、このことを意味する。

Cパート以降では、zとwがパートから得られない場合は、xとyの情報を利用する。「o*」は、このことを意味する。

最終的に、Wがひとつに絞れない場合、および、Wがひとつも得られない場合は、解答不能とする。

表 5 過去問に正解するための検索条件

検索条件の要素	要素	+文型	計
(X)	166	19	185
(X,D)	136	2	138
(X,x)	147	5	152
(X,y)	301	5	306
(X,D,x)	23	0	23
(X,D,y)	93	0	93
(X,x,y)	60	1	61
A パート小計	926	32	958
B パート小計	2	0	2
C パート小計	10	0	10
合計	938	32	970

5 解答にどのパートまで必要か

ここでは、まず、データベースに登録した過去問 970 問を解く場合、どのパートまでで正解を出力できるかを調べた。その結果を表 5 に示す。過去問を解く場合は、正解を導くレコードは必ず存在するため、検索条件を緩めるフィードバックがかかることはない。この表の「要素」の欄は、各パートから取得した要素をすべて検索条件として用いた場合、「+文型」の欄は、それに各パートの文型を追加した場合の、答を一意に絞り込めた問題数を示す。

この表から以下のことがわかる。

1. 過去問の 98.8% (958/970) は、A パートで正解を導くことができる。

データベースのサイズが大きくなると、この値は低下すると考えられるが、実際に出題されるパラレル問題の大半は、事実上、A パートだけで正解が定まる問題となっている可能性がある。

2. 正解を導くのに文型が必要な問題は、3.3% (32/970) にすぎない。

A パートから X 以外の要素が得られた 773 問中、文型が必要だったのは 13 問 (1.7%) のみで、残りの 760 問は抽出した要素だけから答が一意に決定できた。一方、X しか得られなかった場合は、185 問中 19 問 (10.3%) が文型を必要とした。このことは、設定した x, y, D が、答を絞り込むために適切に機能していることを意味する³⁾。実際、先の 760 問中、X だけで答を一意に絞り込める問題は 523 問で、残りの 237 問は X 以外の要素が答の限定に寄与している。

次に、逆方向のレコードを使用する場合と使用しない場合で、正解を出力するために必要なパートが

3) これらの要素が抽出できれば、パート内の文型の差異は無視できる。

表 6 逆方向のレコードの影響

	両方向			順方向のみ		
	要素	+文型	計	要素	+文型	計
A パート	926	32	958	938	27	965
B パート	2	0	2	1	0	1
C パート	10	0	10	4	0	4

どう変化するか調べた。その結果を表 6 に示す。逆方向のレコードを使用しない場合は、A パートのみで正解を出力できる問題が 7 問増える。逆方向レコードを使用すれば、潜在的に解ける問題が 2 倍になることを考慮すれば、この差は許容できる。実際、現在のシステムは、過去問 970 問の前半部と後半部を入れ替えた問題のうち、950 問を解くことができる⁴⁾。

現在のシステムは、解答判定に基づき検索条件を緩めたり強めたりするので、A パートの文型の揺れをある程度許容する。たとえば、以下の例では、いずれか一方が過去問として登録されていれば、他方を解くことができる。

- (2) a. 『半七捕物帳』を書いたのは岡本綺堂ですが、『人形左七捕物帳』を書いたのは誰でしょう？
b. 『半七捕物帳』の作者は岡本綺堂ですが、『人形左七捕物帳』の作者は誰でしょう？

6 おわりに

本研究では、競技クイズのパラレル問題の基本構造と文型を分析し、問題文の途中までで解答するシステムを作成した。現在のシステムは、答を導くための知識源として過去問のみを利用しており、過去問と同一の問題、前半と後半を入れ換えた問題、文型が一部異なる問題を解くことができる。しかしながら、完全に未知の問題に対しては無力である。

競技会に参加する人間の解答者は、過去問を調査するのに加え、出題の可能性のあるパラレル問題を予想し、準備する。この準備は、本システムでは、データベースに新たなレコードを追加することに対応する。パラレル問題の基本構造である 4 項構造において、まず対比関係となるペア (X と Z) を見つけ、それらに対して残りの 2 項 (Y と W) を設定すれば、新しいレコードを作成できる。すなわち、未知の問題を解く能力と、問題を作成する能力は、本質的に同じである。

4) 解けない 20 問は、前半部と後半部の表現の違いが原因である。

参考文献

- [1] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefler, and Chris Welty. Building Watson: An overview of the DeepQA project. *AI Magazine*, Vol. 31, No. 3, pp. 59–79, Jul. 2010.
- [2] 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. Association for Computational Linguistics, 2017.
- [3] Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. SearchQA: A new Q&A dataset augmented with context from a search engine. *CoRR*, 2017.
- [4] Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan L. Boyd-Graber. Quizbowl: The case for incremental question answering. *CoRR*, 2019.
- [5] 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 回年次大会発表論文集, pp. 237–240, 2020.