D-006

競技クイズにおける問題文の別解・ミスリード検出
Detection of Alternative Answers and Misleading Questions in Quick Fire Quiz

河合 弘理[1)]　湯本 高行[1)]
Hiromichi Kawai　Takayuki Yumoto

1　はじめに

クイズの分野の 1 つに競技クイズがある。競技クイズとは、スポーツのように競い合う早押しクイズの一つである。競技クイズの特徴として、競技者が自ら問題の作成を行い、互いに問題を出題し合う点が挙げられる。この作問は想像以上に時間がかかる。1 問 1 問、問いたい内容が伝わるか、情報が正しいか、答えが限定できているかなどの確認が必要になる。クイズの競技性を保証するために、問題に不備がある状況はなるべく避けたい。

そこで本研究では、クイズの問題に対して別解とミスリードを検出するアプローチを提案する。なお、クイズの解答生成においては、すでに非常に高い精度のモデルが完成している [1]。そのため、本研究では既存の解答生成モデルを使用する。まず、別解検出では別解を 2 つに分類し、それぞれ別解で異なる手法を用いて検出を行う。またミスリード検出では、途中までの問題文を解答生成モデルに入力し、生成された解答と正解の固有表現の違いに注目して検出を行う。

2　関連研究

クイズの解答生成モデルの最新の研究として有山らの研究 [1] がある。この論文では、「AI 王 〜クイズ AI 日本一決定戦〜」というコンペティションの第 2 回と第 3 回の結果や分析がまとめられている。このコンペティションでは日本語のクイズ問題を題材とした日本語質問応答データセットを用いて、モデルが生成する解答の正解率を向上させることを目標としている。

第 3 回コンペティションでの結果を見てみると、NEC データサイエンス研究所のチーム「レヴォ」が 94.8%もの高い正解率を記録していた。また株式会社ベルシステム 24 ホールディングスのチーム「ICS Lab.」も 94.4%であった。このように解答生成モデルは既に十分な精度のモデルを作成できる。

3　クイズの問題文の別解・ミスリード検出

本論文内では「正解」と「解答」が使い分けられている。「正解」はあらかじめ決められた問題の答えで、「解答」はモデルにより生成された答えである。また、解答生成モデルで取得できる sequence score の自然対数を求め、生成した解答の「確信度」とする。sequence score と確信度の定義式を式 1、式 2 に示す。なお、確信度は 0 から 100 の値をとる。また、$score_k$ は出力される単語系列における各単語の生成確率である。

$$sequence\_score = \sum_{k=1}^{n} \log\left(score_k\right) \tag{1}$$

$$\text{確信度} = \exp(sequence\_score) \times 100 \tag{2}$$

3.1　問題定義

3.1.1　別解

別解とは想定された正解以外の正解を指す。本研究では別解を以下の 2 種類に分類する。

- 同義的別解
- 異義的別解

「同義的別解」は、辞書をみれば確認できる別名や表記ゆれのことを指す。このような別解はクイズの問題集によっては記載されていることもあるが、中には記載が一切なく困ることがある。同義的別解の例を例 1-1 に示す。

例 1-1) 現在の愛知県に生まれ、のちに江戸幕府を開いて初代将軍になった日本の武将は誰でしょう?
A. 徳川家康

徳川家康は過去に何度か改名しており「竹千代」や「徳川元康」などの別名がある。この例だとどちらも誤答とは言えない。この事を作問者が知らなかった場合、徳川家の別の人物を答えたと思い誤答となってしまう。

また「異義的別解」は別名や表記ゆれとは別に、問題文により発生する別解である。100 文字もない問題文の中に情報を詰め込むのは難しく、浅い裏取りでは問題に対する答えが一意に定まらないことがある。このような「問題文によって発生する別解」を本研究では「異義的別解」と定義する。

異義的別解の例を例 2-1 に示す。

例 2-1) 兵庫県にある山は何でしょう?
A. 六甲山

兵庫県には六甲山だけでなく、氷ノ山などの多くの山が存在する。そこでこれらの山を別解として検出したい。しかしながら、この別解はただ同義的別解を集めるだけでは対応が難しい。

3.1.2　ミスリード

ミスリードは、バラエティのクイズ番組などの「ひっかけ問題」を想像してもらえると分かりやすい。問題文の途中を途中まで読んで溜めを作り、その間に解答者に 1 度解答させた後で、「ですが」と言うような形式の問題だ。こういったひっかけ問題は、意図的なものである。しかし、本研究における「ミスリード」は、作問者の「意図しないひっかけ」が発生してしまうことを指す。例 3-1 を例に考えてみる。

例 3-1) 静岡県と山梨県に跨る、日本一高い山である富士山の標高は何 m でしょう?
A. 3776m

1)　兵庫県立大学大学院情報科学研究科 Graduate School of Information Science, University of Hyogo

全文を読めば答えが「3776m」であることに問題が、早押しクイズとして考えると問題が生じる。例 3-2 は問題が途中まで読まれた時の例である。 なお、「/」はボタンが押されたところを表す記号である。

例 3-2) 静岡県と山梨県に跨る、日本一高い/

ここで押したとき、多くのクイズプレイヤーは「静岡県と山梨県に跨る、日本一高い/山は何でしょう?」と続くと考え「富士山」と答える。この時点まで読まれた情報が「富士山」に付随する情報であったため、そう答えるのが自然だと考えるからである。

このような例が本研究で検出したいミスリードである。意図しないミスリードが発生すると、作問者にとっても解答者にとっても不本意な結果になってしまう。特に競技クイズにおいて、ミスリードは避けるべき事象である。そのため、作問者はミスリードになりうるポイントを理解しておくことが望ましい。

3.2 別解検出

本研究では、同義的別解と異義的別解にそれぞれ異なるアプローチから別解検出を行い、その結果を合わせることで検出精度の向上を目指す。別解検出の流れを図 1 に示す。

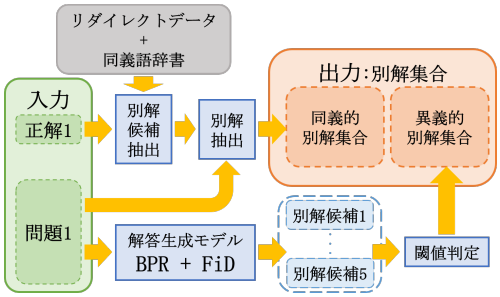

図 1 別解検出の流れ

3.2.1 Wikipedia データからの同義的別解の検出

Wikipedia には記事のタイトルと近い名前や別名で検索すると元のタイトルの記事にリダイレクトする機能がある。本手法ではこのリダイレクト機能を用いる。また、Wikipedia ではダンプデータ[1]を提供されており、その中のリダイレクトデータには、記事のタイトルとその記事にリダイレクトできる別名や表記揺れの情報が入っている。タイトルとリダイレクト先の対応は一対一対応ではなく、複数のリダイレクト元が同じタイトルと対応する「多対一対応」になっている。リダイレクトの例を表 1 に示す。

表 1 リダイレクトの例

タイトル	リダイレクト元
コーヒー	コーヒ
教皇	ローマ教皇
ダウンタウン_(お笑い)	ダウンタウン
コーヒー	ホットコーヒー

まずはリダイレクトデータから別解候補を発見する。リダイレクト先のタイトルのうち、正解と同じ文字列を含むデータ集合を作成する。なお、Wikipedia では同じ語句の記事が存在するときに、ラベル付けすることで混ざらないようにしている。この性質を利用し、タイトルごとにグループ化する際に、同音異義語の語句が混在しないようにした。ここで集まったタイトルごとのリダイレクト元が各問題の別解候補となる。

次に別解候補から別解を選択する。もし正解と同じタイトルを含むデータ集合が 1 つだけだった場合、それがそのまま別解となる。データ集合が複数あった場合、括弧内の言葉に注目する。表 1 の「ダウンタウン_(お笑い)」の例だと「お笑い」に注目し、問題文の中にその言葉がないか調べる。もしあればその言葉がついている別解候補を、なければ括弧がついていない別解候補を別解とする。

さらに本研究では、補助的なツールとして chikkarpy[2]を使い、同義語の候補を追加した。chikkarpy は同義語検索ができる Python のライブラリである。Wikipedia のリダイレクトに比べて余計な単語は少ないため同義的別解の収集に最適であるものの、取得できる単語は多くない。

以上のようにリダイレクトデータと chikkarpy から集めた別解集合を、解答に対する別解として出力する。

3.2.2 解答生成モデルを用いた異義的別解の検出

まずは解答生成モデルから別解候補となる解答を取得する。解答生成には、Binary Passage Retriever(BPR)[3]と Fusion-in-Decoder(FiD)[4] を組み合わせたモデルを使用する FiD モデルではビームサーチ用いて、最大 5 つの解答を生成できるようにした。生成された 5 つの解答のうち、「正解と同じ解答」または「問題文中にある解答」を除いた残りを各問題の別解候補とする。

各別解候補は確信度と共に出力される。本研究ではこの確信度を元に各別解候補から別解抽出を行う。別解の判定基準を表 2 に示す。

表 2 別解候補の基準 ($\theta_1 < \theta_2$)

	候補内に正解を含む	候補内に正解を含まない
全てが θ_1 未満	判断不可	判断不可
全てが θ_2 未満	正解以外は別解 (θ_1 以上)	別解 (θ_1 以上)
θ_2 が存在	別解なし	別解 (θ_1 以上)

解答の確信度が全てが θ_1 未満のものは、別解が存在しているのか、解答生成が正しくできていないのか分からないため、今回は全て除くことにした。残った解答のうち、確信度が 1 つでも θ_2 以上かつ正解を含む出力は、正解のみを正しく出力できているものとして「別解なし」とした。残りの解答のうち θ_1 以上で正解していない解答を別解として出力した。

3.3 固有表現抽出を用いたミスリード検出

ミスリード検出の流れを図 2 に示す。

ミスリード検出でも BPR-FiD モデルを使用して解答を取得する。本手法では生成する解答は 1 つだが、実際に早押しで聞いているような状況を再現するため、入力は問題文全文ではなく、問題文を形態素単位で区切ったものとする。最初は文頭から形態素単位で 4 つ繋げた状態から入力し、問題文が全文に戻るまで形態素を足し合わせながら解答の生成結果を出力する。入力する問題文のイメージを以下に示す。なお、ここでの「/」は形態素

1） https://dumps.wikimedia.org/jawiki/

2） https://github.com/WorksApplications/chikkarpy

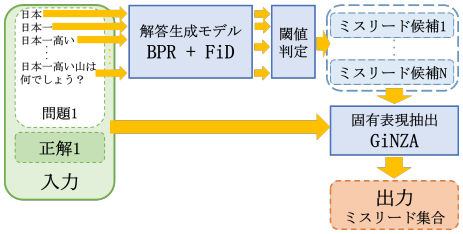

図 2　ミスリード検出の流れ

単位の区切れを示す。

日本/一/高い/山/
日本/一/高い/山/は/
⋮
日本/一/高い/山/は/何/で/しょう/?/

この出力結果からミスリード候補を取得する。候補の決定には確信度を使用する。本研究では生成した 5 つの解答のうち確信度がもっとも高い解答を使用し、その確信度が α_1 以上あり、直後の解答の確信度が α_2 以上低くなる解答をミスリード候補とした。次に各ミスリード候補と正解データに対して固有表現抽出を行う。本研究では「問題文+正解」と「途中までの問題文+ミスリード候補」の 2 つの文章を作成し、ミスリード候補と正解データについた固有表現を取得する。取得するミスリード候補と正解データの固有表現の例を表 3 に示す。

表 3　固有表現抽出の出力例

正解	ミスリード候補	固有表現 (正解)	固有表現 (ミスリード候補)
江戸川乱歩賞	東野圭吾	Award	Person
1492 年	レコンキスタ	Date	Continental_Region
25 度	冬日	Temperature	Date

固有表現抽出には GiNZA[3] のモデルの 1 つである、Transformers 事前学習モデルを用いた ja_ginza_electra を使用する。ja_ginza_electra は、従来の ja_ginza モデルと比較すると処理速度は遅いが、解析精度が向上しているモデルである。これにより付与したミスリード候補と正解データの固有表現を比較し、異なっているものをミスリードとして検出する。

4　実験

4.1　使用したデータ

4.1.1　使用した解答生成のモデル

解答生成には BPR-FiD モデルを使用する。BPR は東北大学自然言語処理チームが公開している、クイズ AI 王の問題に Wikipedia のパッセージを付与したデータセットで学習したモデル [6] を使用する。このデータセットは、第 3 回クイズ AI 王のコンペティション (2022 年 8 月〜2022 年 12 月開催)[4] で使用された 22335 問のクイズ問題に対して、問題と関連度が高い Wikipedia 記事のパッセージを付与することで作成された。クイズ問題は「abc/EQIDEN」というクイズ大会の第 1 回（2003 年）から第 12 回大会（2014 年）で使用された問題である。また Wikipedia の記事は 2022 年 4 月 4 日時点のデータを使用している。BPR 内で使用されている BERT は東北大学が公開している日本語の事前学習済みモデル[5] である。FiD は、第 4 回クイズ AI 王のコンペティション (2023 年 10 月〜2024 年 1 月開催)[6] で公開されているモデルを使用する。このモデルは日本語コーパス用いて事前学習を行った T5 モデル [7][7] を使用している。

この解答生成モデルの精度を調べる。問題集『あなたがクイズを始めるための本　part1』[8] の全問題 1000 問に対して、生成する 5 つの解答のうち正解ラベルが含まれているものを正解とする。その結果 1000 問中 809 問で正解を生成することができた。

4.1.2　別解検出のデータセット

別解検出であらかじめ決まった別解が存在する問題と別解が想定されていないデータを使用する。今回は Q-groups が作成した問題集『あなたがクイズを始めるための本』[9][10] シリーズ 3 冊と、『あなたがクイズに強くなるため本　part1』[11] の計 4 冊から別解が用意されている問題データ 156 問と別解が用意されていない問題 156 問の合計 312 問を評価データとした。データの例を表 4 に示す。なお、別解ラベルは問題集に記載の別解を指す。

表 4　別解検出の問題データ

問題 (正解, 別解ラベル)
問題 1：正式には「西瀬戸自動車道」という、広島県尾道市と愛媛県今治市を結ぶ道路の通称は何でしょう? (正解：しまなみ海道, ラベル：瀬戸内しまなみ海道)
問題 2：世の中にはいろんな権利がありますが、単に「有権者」という場合に保有しているのは何の権利でしょう? (正解：参政権, ラベル：選挙権、投票権)
問題 3：釣り用語で、魚が一匹も釣れなかったことを何というでしょう? (正解：ボウズ)

4.1.3　ミスリード検出のデータセット

ミスリード検出では、別解検出と同じ問題集 4 冊を参考に私が作成したミスリードが想定される問題 71 問と、問題集からランダムに選んだ問題 73 問の合計 144 問を評価データとした。データの例を表 5 に示す。なお、ミスリードラベルは想定されるミスリードを指す。

4.2　別解判定の評価

4.2.1　実験方法

本研究では異義的別解の閾値の基準を $\theta_1 = 70\%, \theta_2 = 80\%$ として、同義的別解、異義的別解、2 つの和集合を取った「組み合わせ」の 3 つを対象に評価を行う。評価項目には検出率と妥当性を用いる。

検出率　検出率は正解の有無に関係なく別解が検出できたかどうかの 2 値分類で評価を行う。分類の評価指標となる適合率、再現率の定義式を式 3〜4 に示す。適合率は検出された別解の正しさ、再現率は検出漏れの少なさを表す。

3）https://megagonlabs.github.io/ginza/
4）https://sites.google.com/view/project-aio/competition3?authuser=0
5）https://huggingface.co/cl-tohoku/bert-base-japanese-v3
6）https://sites.google.com/view/project-aio/competition4?authuser=0
7）https://huggingface.co/sonoisa/t5-base-japanese
8）https://q-tak.com/?p=4916
9）https://q-tak.com/?p=4998
10）https://q-tak.com/?p=6476
11）https://q-tak.com/?p=4918

表 5　ミスリード検出の問題データ

問題 (正解, ミスリードラベル)
問題 1：ヨーロッパに似た町並みから「南米のパリ」という別名があるブエノスアイレスは、どこの国の首都でしょう? (正解：アルゼンチン, ラベル：ブエノスアイレス)
問題 2：伝教大師という称号でも知られる最澄によって開山された、天台宗の総本山である滋賀県の寺院はどこでしょう? (正解：最澄, ラベル：延暦寺)
問題 3：抗酸化作用があることでも知られる、柿やトマトに含まれる赤色の色素は何でしょう? (正解：リコピン)

$$\text{適合率} = \frac{\text{検出された別解ラベル付きデータの数}}{\text{検出されたデータの数}} \quad (3)$$

$$\text{再現率} = \frac{\text{検出された別解ラベル付きデータの数}}{\text{別解ラベル付きデータの数}} \quad (4)$$

また、F 値は適合率と再現率の調和平均である。

妥当性　出力された解答を見ると、想定されていないが別解になりうる出力が多く存在していた。そこで、正解データを無視して出力した別解がどれぐらい妥当かを調べる。これを妥当性として定義した。定義式を式 5 に示す。

$$\frac{1}{N}\sum_{i=1}^{N}\frac{\text{問題}_i\text{で検出した別解のうち妥当な別解数}}{\text{問題}_i\text{で検出した別解数}} \quad (5)$$

出力された別解が妥当かどうかは私自身で判別を行った。今回は以下の 4 つの基準に従って判断を行った。

1. 問題文による限定がない
2. 問いに対する解答の対象として正しい
3. 間違いと言い切れない
4. ひらがな・漢字・カタカナ・外国語の違い

適合率の修正　検出率の節で定義した適合率は、別解として検出したデータのうち、別解ラベル付きデータがどの程度あるかを示していた。しかし、別解ラベルが付与されていないデータにも妥当な別解が存在している。そこで、式 4 を検出した別解がどれほど妥当だったかを新たに適合率として定義する。修正後の適合率の定義式を式 6 に示す。

$$\text{適合率 (修正後)} = \frac{\text{妥当な別解を検出したデータ数}}{\text{別解を検出したデータ数}} \quad (6)$$

4.2.2　実験結果

検出率　各手法における別解の検出数は表 6〜8 のようになった。

表 6　同義的別解の検出結果

	別解を検出	別解の検出なし	合計
別解あり	87	69	156
別解なし	90	66	156
合計	177	135	312

表 7　異義的別解の検出結果

	別解を検出	別解の検出なし	合計
別解あり	61	95	156
別解なし	27	129	156
合計	88	224	312

表 8　組み合わせの検出結果

	別解を検出	別解の検出なし	合計
別解あり	128	28	156
別解なし	105	51	156
合計	233	79	312

手法ごとの評価指標の割合は表 9 のようになった。

表 9　各手法の評価

	同義的別解	異義的別解	組み合わせ
適合率	0.4915	0.6932	0.5494
再現率	0.5577	0.3910	0.8205
F 値	0.5225	0.5000	0.6581

絶対的な精度はそこまで高くないものの、同義的別解と異義的別解のどちらかより組み合わせた方が、相対的にみると精度が高くなることが分かる。

妥当性　設定した基準に沿って妥当性を求めた結果、同義的別解では 59.21%、異義的別解では 61.74%、組み合わせた手法では 59.59%と、どの手法も約 60%程度の妥当性があることが分かる。

適合率の修正　修正後の適合率の分析結果を表 10 に示す。

表 10　各手法の適合率 (修正後) $\theta_1 = 70\%$

	妥当な解答数	検出数	適合率 (修正後)
同義的別解	136	177	0.7684
異義的別解	57	88	0.6705
組み合わせ	178	233	0.7639

実験後の結果を前提としているため正しい指標とは言えないが、より適切な別解ラベルがあれば、適合率が高くなることが期待できる。

閾値　本研究では、θ_2 は 80%で固定し、θ_1 の値を 50%, 60%, 70%と変化させて閾値の調整を行った。それぞれの閾値で式 6 で定義した適合率の結果についての検討を表 11, 12 に示す。

表 11　異義的別解の適合率 (修正後)

閾値	妥当な解答数	検出数	適合率 (修正後)
$\theta_1 = 50\%$	64	118	0.5424
$\theta_1 = 60\%$	60	101	0.5941
$\theta_1 = 70\%$	57	88	0.6705

表 12　組み合わせの適合率 (修正後)

閾値	妥当な解答数	検出数	適合率 (修正後)
$\theta_1 = 50\%$	181	248	0.7298
$\theta_1 = 60\%$	179	239	0.7490
$\theta_1 = 70\%$	178	233	0.7639

どちらの手法も閾値の値を高くすると適合率 (修正後) も高くなった。また、検出数が少なくなっても妥当

な解答数の変化はほぼなかった。従って今回は閾値を $\theta_1 = 70\%$ にして評価を行った。

4.2.3 別解の出力結果についての分析

別解のうまくいった出力例を表 13 に示す。問題 1 は同義的別解の検出がうまく機能した例、問題 2 は異義的別解の検出がうまく機能した例である。

表 13 うまくいった検出例

問題 (正解, 別解ラベル, 同義的別解, 異義的別解)
問題 1：中国・唐の時代の詩人で、玄宗皇帝と楊貴妃の愛を綴った叙事詩『長恨歌』で知られるのは誰でしょう? (正解：白居易, ラベル：白楽天, 同義的別解：白楽天)
問題 3：サッカーゲーム『ウイニングイレブン』や野球ゲーム『実況パワフルプロ野球』で知られるゲーム会社はどこでしょう? (正解：コナミ, 異義的別解：コナミデジタルエンタテインメント)

次にうまくいかなかった検出例を表 14 に示す。問題 1 は同義的別解が問題文が問う対象とは異なっていた例である。また、問題 2 は異義的別解がうまく機能しなかった例である。

表 14 うまくいかなかった検出例

問題 (正解, 別解ラベル, 同義的別解, 異義的別解)
問題 1：実写映画『ちはやふる』で、ヒロインのカルタ取り：綾瀬千早を演じた女優は誰でしょう? (正解：広瀬すず, 別解ラベル：久家心, 同義的別解：17 才のすずぼん)
問題 2：サンショウの木で作られたものがよいとされる、すり鉢でものをすりつぶすのに使われる棒のことを何というでしょう? (正解：すりこぎ, 異義的別解：当たり棒)

4.2.4 考察

同義的別解と異義的別解を組み合わせることで精度の良い検出が可能だと考えられる。表 13 の問題 2 のように元々別解がなかったが、別解として妥当な解答が存在したため、この検出はある程度有効であるといえる。

さらに精度を上げるに、まず同義的別解を検討する。現状では、Wikipedia のリダイレクトデータには記事内のデータや間違い入力も混在しており、これが精度低下の原因の 1 つだと考えている。また、クイズでは正式名称でないが一般となっている名称や別名を正解とする場合もあり、リダイレクト先とリダイレクト元のどちらを用いるべきかは一概には決められない。

また異義的別解は、ベースとしている解答生成の知識ベースを更に充実させたり、クイズ AI 王でもあったように BM25 を導入して関連文書をリランキングさせることでより精度を高くすることが可能であると考える。

加えて、異義的別解の閾値も暫定的なものなので、再度調べる必要があると考える。妥当性のある別解集合を検出することはできるが、具体的にどれが別解といえるかを判断するのに労力がかかってしまうので、そこをどうするかという課題も残る。

4.3 ミスリード検出の評価

4.3.1 実験方法

本研究ではミスリード候補の基準となる閾値を $\alpha_1 = 90\%, \alpha_2 = 5\%$ として評価を行った。評価にはミスリードの有無の適合率、再現率、F 値と解答の再現率と検出正解率を用いる。

ミスリードの有無の適合率・再現率・F 値 まずは別解の時と同じく、正解の有無に関係なくミスリード検出の有無で 2 値分類で評価を行う。評価指標となる適合率、再現率の定義式を式 7〜8 に示す。

$$\text{適合率} = \frac{\text{検出されたラベル付きデータの数}}{\text{ミスリードを検出した問題データの数}} \tag{7}$$

$$\text{再現率} = \frac{\text{検出されたラベル付きデータの数}}{\text{ラベル付きデータの数}} \tag{8}$$

また、F 値は適合率と再現率の調和平均である。

解答の再現率 解答の再現率は、ミスリードラベルのあるデータのうち実際に正しく検出できた割合を表している。解答の再現率の定義式を式 9 に示す。

$$\frac{\text{ラベルと同じ検出をしたデータ数}}{\text{ラベルのデータ数}} \tag{9}$$

検出正解率 最後に検出したミスリードのうち正しく検出できた割合を調べる。「検出正解率」の定義式を式 10 に示す。

$$\frac{\text{ラベルと同じ検出をしたデータの数}}{\text{検出したラベル付きデータ}} \tag{10}$$

閾値 ミスリード候補の取得の際に設定した、閾値の α_1 と α_2 を調整する。本研究では $(\alpha_1, \alpha_2) = (80\%, 10\%), (80\%, 5\%), (90\%, 5\%)$ の 3 つの場合の「正解率」と「検出正解率」の評価を行う。

4.3.2 実験結果

ミスリードの有無の適合率・再現率・F 値 検出されたミスリードの問題データの数は表 15 のようになった。

表 15 ミスリード検出における検出結果

	ミスリードを検出	ミスリードの検出なし	合計
ミスリードラベルあり	35	36	71
ミスリードラベルなし	12	61	73
合計	47	97	144

適合率は 74.47%、再現率は 49.30%、F 値は 59.32%であった。

解答の再現率 ミスリードラベルを付与した 71 件のデータのうち、正しく検出できたデータは 30 件で、ミスリード解答の再現率は約 42%と小さい値となった。

検出正解率 ミスリードラベルが付与されたデータのうち検出できたデータは 35 件あり、そのうちミスリードラベルと同じ結果を検出できたデータは 30 件あった。よって検出正解率は約 86%であった。

閾値 閾値ごとの「ミスリード解答の再現率」と「検出正解率」を表 16 に示す。

表 16 各閾値ごとの解答の再現率と検出正解率

閾値	解答の再現率	検出正解率
$\alpha_1 = 80\%, \alpha_2 = 10\%$	25/71 = 0.3521	25/36 = 0.6944
$\alpha_1 = 80\%, \alpha_2 = 5\%$	33/71 = 0.4648	33/43 = 0.7674
$\alpha_1 = 90\%, \alpha_2 = 5\%$	30/71 = 0.4225	30/35 = 0.8571

α_1 を上げていったところ、検出数は少し下がったが誤検出が減り、検出正解率も向上した。また、α_2 を 10%

から 5%にしたところ正しく検出できたデータ数が増えた。そのため、本研究では $\alpha_1 = 90\%, \alpha_2 = 5\%$ を閾値に設定した。

4.4 ミスリードの出力結果

ミスリードのうまくいった出力例を表 17 に示す。ここでの「/」は問題文中にミスリードを検出したタイミングを表している。

表 17 うまくいった出力例

問題 (正解, ミスリードラベル, ミスリード)
問題：ヨーロッパに似た町並みから「南米のパリ」という別名がある/ブエノスアイレスは、どこの国の首都でしょう? (正解：アルゼンチン, ラベル：ブエノスアイレス, ミスリード：ブエノスアイレス)

次にうまくいかなかった出力例を表 18 に示す。

表 18 うまくいかなかった出力例

問題 (正解, ミスリードラベル, ミスリード)
問題 1：百人一首の在原業平の和歌に由来するタイトルをもつ末次由紀/の漫画ちはやふるが題材としている競技は何でしょう? (正解：かるた, ラベル：ちはやふる, ミスリード：伊勢物語)
問題 2：1860 年 3 月、大老・井伊/直弼が浪士の一党によって暗殺された事件を、この事件が起きた場所から何というでしょう? (正解：桜田門外の変, ミスリード：井伊直弼)

問題 1 は全く異なるものをミスリードとして検出している例である。また問題 2 では問題文中にすでに出た単語を繰り返している。

4.4.1 考察

ミスリードの検出率が低かった原因を探るため、ミスリード候補の段階で調べる。ここでは、ミスリード候補の中から人力で、想定されていたミスリードと同じ解答を含むものを抽出した。分析結果はミスリード解答の再現率が約 0.6197、検出正解率が約 0.8627 であった。

ミスリード候補の段階では検出できているがミスリードとして検出できていない問題が 14 問存在した。そこで、ミスリード候補にはあったもののミスリードとして出力できなかった例を表 19 に示す。

表 19 ミスリード候補から抽出できなかった例

問題 (正解, ミスリードラベル, ミスリード)
問題 1：援助や配慮を必要としている人が配慮の必要性を知らせるために提示するヘルプマークに描かれている 2 つの図形とは十字と何でしょう? (正解：ハート, ラベル：ヘルプマーク, ミスリード：ヘルプマーク)
問題 2：代表作に銀河鉄道 999 や宇宙戦艦ヤマトがある漫画家・松本零士と同じ日に生まれた、代表作にサイボーグ 009 や仮面ライダーがある漫画家は誰でしょう? (正解：石ノ森章太郎, ラベル: 松本零士, ミスリード：松本零士)

抽出できなかった例のほとんどは「固有表現が付与できなかった」というパターンであった。また、問題 2 のように正しくラベリングされたが、正解ラベルと別解ラベルが同じ固有表現を持つために検出されなかったものが 2 件あった。生成確率が他の候補より低くて間違ったものが検出されたものは、2 件のみであった。

以上のことから、ラベリングの精度を向上させればミスリード検出として使える可能性は残っていると考える。ラベリングの手法は再考する必要がある。

更なる精度向上の手法としては、「固有表現抽出モデルを学習させて、より多くの解答に対してラベリングを行う」「文章生成系のモデルを使用して途中までになっている問題文を完成させ、固有表現抽出の精度を向上させる」などが考えられる。

5 おわりに

本論文ではクイズの解答生成モデルを利用した別解検出およびミスリード検出の手法を提案した。別解検出では、同義的別解と異義的別解で異なるアプローチを行い、その結果を組み合わせた。ミスリード検出では解答生成モデルに形態素単位で入力し、その結果と確信度の変化を元にミスリードの検出を行った。

別解検出の結果は、2 つの手法を組み合わせた方が精度が高くなることを確認することができた。ミスリード検出は、まだ F 値は約 59%と精度に課題は残るが、ミスリード候補の段階をみるとまだ改善の余地があるため、精度向上は期待できる。

今後の課題としては、やはり精度向上面での課題があると考えられる。また、本研究では解答生成モデルそのものはベースラインをそのまま利用したが、精度の高い解答生成モデルでも再度調べる必要がある。

謝辞

本研究は JSPS 科学研究費助成事業 24K15195 による助成を受けたものです。ここに記して謝意を表します。

参考文献

[1] 有山 知希, 鈴木 潤, 鈴木 正敏, 田中 涼太, 赤間 怜奈, 西田 京介. “クイズコンペティションの結果分析から見た日本語質問応答の到達点と課題”, 自然言語処理, 2024, 31 巻, 1 号, pp.47–78, 2024

[2] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. “Dense Passage Retrieval for Open-Domain Question Answering”, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.6769–6781, 2020

[3] Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. “Efficient Passage Retrieval with Hashing for Open-domain Question Answering”, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp.979–986, 2021

[4] Gautier Izacard and Edouard Grave. “Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering”, Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp.874–880, 2021

[5] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. “Improving Language Understanding by Generative Pre-Training”, Open AI Technical Report, 2018

[6] 山田育矢, 鈴木正敏, 山田康輔, 李凌寒. “大規模言語モデル入門”, 技術評論社, pp.275-304, 2023

[7] 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.” arXiv preprint arXiv:1910.10683. 2019