カタカナ語の意味分類に対する大規模言語モデルの有効性検証

小滝 主紀 (茨城大学大学院 理工学研究科) †
佐々木 稔 (茨城大学 工学部)

'Validation of a Large-Scale Linguistic Model for Semantic Classification of Katakana Words'

Kazuki Kodaki (Graduate School of Science and Engineering, Ibaraki University)
Minoru Sasaki (College of Engineering, Ibaraki University)

要旨

本稿では、LLM を用いてカタカナ語の文脈中の意味分類を行った手法と結果について報告する。ChatGPT などの生成 AI の学習に用いられる資源の多くは英語で占められており、日本語の資源はあまり使用されていない。そのため日本語に含まれるカタカナ語は対応する英単語の意味と異なる場合があり、文脈中の意味分類が正しく行われない可能性が高い。そこで『現代日本語書き言葉均衡コーパス』(BCCWJ) に含まれる文章からカタカナ語を含む文章を抽出し、その中から数個の単語を対象として、gpt-3.5-turbo, gpt-4o, gpt-4o-mini, Gemini-Pro, Swallow の 5 つの LLM を用いて Few-shot Learning を行った。実験 1 と実験 2 では生成 AI が作成した意味区分を利用した際の意味分類とプロンプト中で役割を与えることによる影響を、実験 3 では人間の定義した Wiktionary の意味区分を利用した場合の意味分類を上記の LLM で検証した。結果として生成 AI、Wiktionary どちらの意味区分を扱った意味分類でも gpt-4o が最も平均正解率が高く、gpt-4o と Gemini-Pro は役割を与えることでほとんどのプロンプトで回答精度が向上したことが確認できた。また gpt-4o-mini と Gemini-Pro では Wiktionary の意味区分を利用したほうが平均正解率が 20% 以上高くなった。さらに単語による各 LLM 間での正解率の差異もみられ、gpt-4o,gpt-4o-mini,Gemini-Pro 間で顕著であった。

1. はじめに

LLM(大規模言語モデル) の発展は急速なものであり、検証の余地が多くある。特に自然言語処理分野において研究が推し進められており、日本においても LLM の研究は盛んである。最近ではサイバーエージェント社の『CyberAgentLM3』や、rinna 社の『Youko』など日本語に特化した LLM も開発、公開されている。しかし、生成 AI の中でも使用率の高い ChatGPT や Gemini などは英語ベースで作成されており、LLM 内で占める日本語データの割合は低いと考えられる。さらに日本語の中でも資源の少ないカタカナ語では、対応する英単語の意味と異なる場合があり、文脈中の意味分類が正しく行われない可能性が高い。そこで本稿では、『現

† 24nm724g@vc.ibaraki.ac.jp

代日本語書き言葉均衡コーパス (BCCWJ)』(1) からカタカナ語を含む文章を抽出し、その中から数個のカタカナ語を対象としてプロンプトを作成し、gpt-3.5-turbo, gpt-4o, gpt-4o-mini, Gemini-Pro, Swallow の 5 つの LLM を用いて Few-shot Learning を行うこととした。研究の目的は、各 LLM でのカタカナ語の意味分類に対する有効性の検証を行い、単語や LLM による回答精度の差異や、現状のカタカナ語に関する LLM の回答精度を確認することである。

2. 関連研究

2.1 大規模言語モデルを用いた語義曖昧性解消

機械学習に基づく語義曖昧性解消の研究には、大きく分けて教師あり学習手法と知識ベース手法の 2 つのアプローチがある。教師あり学習手法は、人間が対象となる多義語に正しい語義のラベルを付けたコーパスによって学習した語義分類モデルを用いて、曖昧な単語に対して適切な意味を分類する。近年の数多くの教師あり語義曖昧性解消モデルでは BERT(2) や RoBERTa(3) といった事前学習された言語モデルを用いて単語や文をベクトル化して語義の識別を行っている (4, 5)。知識ベース手法は語義ラベル付きコーパスを使わず、辞書やオントロジーといった外部知識を用いた分類手法である。単語の語義定義文をベクトル化して語義間の関係を学習する手法 (6) や単語間の類義関係を用いて効果的な語義のベクトルを求める手法 (7) などがある。また、最近では語義曖昧性解消において ChatGPT や Gemini といった大規模言語モデルを用いた生成 AI を利用する試みも検討されている (8, 9)。評価データによる実験において高い性能を示す結果が得られているが、最先端のモデルが達成するレベルにはまだ達していない。

3. 実験

3.1 事前準備

まずはじめに本研究を行うにあたって、『現代日本語書き言葉均衡コーパス (BCCWJ)』からカタカナ語を含む文章の抽出を行った。方法としては、BCCWJ 内の Core_SUW.txt において、”外”とラベルがつけられた単語を含む文章が記述されているファイル名を取得し、そのファイル名からパスを指定して C-XML 中の VARIABLE に含まれるカタカナ語を含む文章を抽出した。

実験で利用するデータは、抽出した各カタカナ語の出現頻度を求め、頻度が 5 以上の単語を含む文章で、さらに単語の意味が異なる文章が 4 つ以上あるものとした。以上の条件から選択した対象単語は [”ポイント”,”カット”,”ホーム”,”セット”,”モデル”,”アップ”] の 6 つである。その後、データ中の各文章を ChatGPT に与え、Few-shot Learning を行うために必要な、単語の意味と単語の意味を解説する補助的な文章を生成させた。例として”カット”という単語を含む文の補助的な文章を作成したプロンプトを以下に示す。

プロンプトの例

”いっそ ショートカットに したいけど、維持するの 大変よ〜！ って言われたので、ガマンガマン （ー”ー¥）”
以上の文章中のカットの意味を文脈に沿って説明してください。

回答が期待した内容であればそのまま利用し、期待した出力がされなかった場合は内容を人手で整形し、文脈に沿った説明文章を作成した。例文では

回答の例

”意味: ここでの「ショートカット」は、髪型を短く切ることを指します。髪を短くすることを望んでいるが、維持が大変なので我慢しているという意味です。”

という説明文章が出力され、これは期待した内容でありテンプレートに沿って形を変え、プロンプト内で利用した。結果として、”カット”を含む例文に関する実験で OpenAI API の LLM に渡すプロンプトは以下のようになった。

完成したプロンプトの一部

prompt = [
...
”role”: ”user”, ”content”: ”3. いっそ ショートカットに したいけど、維持するの 大変よ〜！ って言われたので、ガマンガマン （ー”ー¥）”,
”role”: ”assistant”, ”content”: ”3. のカットは、「髪を切る」。髪を短くすることを望んでいるが、維持が大変なので我慢しているという意味です。”,
...
”role”: ”user”, ”content”: ”仕事帰りに 髪を 切ってきました〜！はい １，０００円カット！。この文章中のカットの意味を推察し、文章中での意味として最も近いものを 1.〜4. の中から選択してください。回答は絶対に数字のみでお願いします。”
]

以上の方法で作成したデータを、実験 1、実験 2 で Few-shot Learning を行う際に活用した。

3.2 実行環境

本実験を行った実行環境と各 LLM について以下に示す。

・実行環境

python 3.11.7

・LLM

OpenAI API

-gpt-3.5-turbo-0125

-gpt-4o-mini

-gpt-4o

Gemini-Pro

Swallow

3.3 実験方法

以下に各 LLM で行った実験 1、実験 2、実験 3 の方法を示す。実験の対象単語は事前準備で述べた 6 つの単語であり、説明文章は各 LLM で記述内容は等しく、実験 1 では特に役割を与えず、実験 2 では言語分析家を与えている。実験 1 から実験 3 では、各 LLM でプロンプトに対する回答を選択肢の中からを 100 回出力させ、選択肢の出力頻度を記録し実行結果とした。そして実験 1、実験 2 では、”ポイント”、”アップ”に関しては選択肢は 5 個、”カット”、”ホーム”、”セット”、”モデル”は選択肢は 4 個で、実験 3 では、”セット”は 8 個、”アップ”は 6 個、”ポイント”、”カット”は 5 個、”ホーム”、”モデル”は 4 個であり、条件を揃えるため各 LLM で temperature は 0.9 に指定している。なお以下で示す各 LLM のプロンプトのテンプレートは実験 2 のものである。

3.3.1 OpenAI API

まず OpenAI API を用いた LLM である gpt-3.5.-turbo、gpt-4o-mini、そして gpt-4o に関して説明を行う。プロンプトのテンプレートとしては、以下の図 1 のようになる。対象単語を含む各文章と、事前準備で作成した文章中の単語の意味と補助的な説明文章をプロンプトとして渡し、テンプレートに沿ってテストデータと選択肢を与え Few-shot Learning を行った。

”回答は絶対に数字のみでお願いします”と記述したのは、出力結果を簡単にするためである。Gemini-Pro と Swallow もこれに倣い記述している。なおプログラムの実行は Anaconda で行った。

```
prompt = [
    #単語,正解
    {"role": "system", "content":"You are a linguistic analyst."},

    {"role": "user", "content": "1.文章"},
    {"role": "assistant", "content": "1.の単語は、「意味」。説明文章"},
    {"role": "user", "content": "2.文章"},
    {"role": "assistant", "content": "2.の単語は、「意味」。説明文章"},
    {"role": "user", "content": "3.文章"},
    {"role": "assistant", "content": "3.の単語は、「意味」。説明文章"},
    {"role": "user", "content": "4.文章"},
    {"role": "assistant", "content": "4.の単語は、「意味」。説明文章"},


    {"role": "user", "content": "テスト文章。この文章中の単語の意味を推察し、\
    文章中での意味として最も近いものを1.〜4.の中から選択してください。回答は絶対に数字のみでお願いします。"}
]
```

図 1　OpenAI API におけるプロンプトテンプレート

3.3.2 Gemini-Pro

次に Gemini-Pro を用いた場合に関して説明を行う。プロンプトのテンプレートとしては、以下の図 2 のようになる。図 2 のように対象単語を含む各文章をはじめに明示し、その後事前準備で作成した文章中の単語の意味と補助的な説明文章をプロンプトとして渡し、テンプレートに沿ってテストデータと選択肢を与え Few-shot Learning を行った。OpenAI API のように role 等は設定していない。なおプログラムの実行は Google Colaboratory を利用した。

```
prompt = [
    #単語,正解
            "あなたは言語分析家です。\
              1.文章\
              2.文章\
              3.文章\
              4.文章\
            以下は1.～4.の文中の単語の意味を説明したものです。\
              1.の「単語」は、「意味」。説明文章。\
              2.の「単語」は、「意味」。説明文章。\
              3.の「単語」は、「意味」。説明文章。\
              4.の「単語」は、「意味」。説明文章。\",

            "テスト文章。\
            この文章中の単語の意味を推察し、文章中での意味として最も近いものを1.～4.の中から選択してください。\
            回答は絶対に数字1文字のみでお願いします。"
]
```

図 2　Gemini-Pro におけるプロンプトテンプレート

3.3.3　Swallow

最後に Swallow を用いた場合に関して説明を行う。プロンプトのテンプレートとしては、以下の図 3 のようになる。図 3 のように対象単語を含む各文章をはじめに明示し、その後事前準備で作成した文章中の単語の意味と補助的な説明文章をプロンプトとして渡し、テンプレートに沿ってテストデータと選択肢を与え Few-shot Learning を行った。OpenAI API のように role 等は設定していない。なおプログラムの実行はクラウドサービスである Modal を利用した。

```
prompt = [
    #単語、正解
            "あなたは言語分析家です。\
            1.文章\
            2.文章\
            3.文章\
            4.文章\
            以下は1.～4.の文中の単語の意味を説明したものです。\
            1.の「単語」は、「意味」。説明文章。\
            2.の「単語」は、「意味」。説明文章。\
            3.の「単語」は、「意味」。説明文章。\
            4.の「単語」は、「意味」。説明文章。\
            以上を踏まえたうえで1.～4.の文章を選択肢として以下の質問に詳細に回答してください。\
            テスト文章。\
            この文章中の単語の意味を推察し、文章中での意味として最も近いものを1.～4.の中から選択してください。\
            回答は絶対に数字1文字のみでお願いします。"
]
```

図 3　Swallow におけるプロンプトテンプレート

3.3.4　意味区分

本セクションでは実験で使用した意味区分を示す。はじめに実験 1、実験 2 で使用した ChatGPT に作成させた各対象単語の意味区分は以下のとおりである。

・ポイント...1. 金融や経済の専門用語 2. 商品やサービスにおける得点 3. 重要な要素 4. 競技やゲームにおける得点 5. 場所

・カット...1. 物理的に切る 2. 野球においてバットでボールを切るように打つこと 3. 髪を切る 4. イベント開始時の儀式

・ホーム...1. 野球用語のホームプレート 2. 駅のプラットホーム 3. ホームセンターの一部 4. 老人ホームの一部

・セット...1. 独立した単位やグループ 2. 背景や装飾 3. 配置 4. 舞台装置

・モデル...1. 職業 2. 製品のバージョンやバリエーション 3. 作品のインスピレーションとなる人物 4. ビジネスや経済における概念や構造

・アップ...1. バスケットボールのプレーの一つ 2. 拡大 3. コンテンツの投稿 4. 向上 5. 列挙

次に実験 3 で使用した Wikitionary で定義された意味区分は以下のとおりである。

・ポイント...1. ドット (点) 2. 場所・地点 3. 要点 4. 得点 5. 尖頭器 (武器の先端や尖った部分)

・カット...1. 切断、又は、その行為により切り取られたもの 2. 削減すること 3. 挿絵などに使われる小品の絵画 4. 映像作品において場面などを切り取ること、切り取った部分又、切り取ったものを組み合わせ編集すること 5. 映画監督などが、撮影を止める際に発する声

・ホーム...1. 家、家庭、故郷 2. 本拠地 3. 老人ホームなどの略 4. プラットホームの略

・セット...1. 一組、一揃い、一対 2. スカッシュ・テニス・卓球・バレーボールなどの試合の一区切り 3. 映画・テレビの撮影や演劇の舞台に用いる、建物や街並みなどを模した設備、舞台装置 4. 機械を設定すること、用意・準備や配置をすること 5. 髪を整えること 6. 集合 7. 受信機、受像機 8. 猟犬が伏せの動作により獲物を指示すること

・モデル...1. 模型、原型、方式、型、雛型 2. 模範、手本 3. 小説などで作品の題材にされた人や物 4. ファッションモデル

・アップ...1. 増加、上昇 2. アップロード 3. クローズアップ 4. クランクアップ、完了、仕上げ 5. ウォーミングアップ 6. アップスタイル、アップヘア

ホームを例に実際に実験で利用したプロンプト中の文章を以下に示す。先に ChatGPT で作成した意味区分について示す。なお以下で示す文章は BCCWJ から抽出したものである。

各文章

1. ３回表 大松が代わったばかりの石川から右中間スタンドに運び、初出場初ホームラン。

2. ホームまで三人に見送りを受けて新幹線に乗ってから、志津は先刻フッと正樹が漏らした言葉を思い返していた。

3. 帰りにホームセンターで、チャリンコに積んでおく用の６角レンチセット３９８円購入。

4. ○○○は、介護予防特定施設入居者生活介護（介護保険のサービス）を行なう有料老人ホームです。

各説明文章

1. の「ホーム」は、「野球用語のホームプレート」。大松選手が初めてホームランを打ち、バッターが全ての塁を回って最後にホームプレートに戻ることを意味しています。

2. の「ホーム」は、「駅のプラットホーム」。新幹線の駅で列車に乗るためのプラットホーム

のことを指します。

3. の「ホーム」は、「ホームセンターの一部」。「ホームセンター」は、家庭用品や DIY 用品を販売する大型店を指します。

4. の「ホーム」は、「老人ホームの一部」。「老人ホーム」は、高齢者が生活し、介護や医療などのサービスを受けられる施設を指します。

テストデータ：改札口とホームへの階段以外にまともな照明がないのです。

次に Wiktionary で定義された意味区分について示す。こちらで示す文章は ChatGPT に作成させたものである。

各文章

1. 長い旅行から帰って、やっとホームに戻った感じがします。

2. 私たちのチームは来週、ホームで重要な試合を行います。

3. 彼のお祖母さんは市内の老人ホームに住んでいます。

4. 電車がホームに到着したので、乗客たちは急いで乗り込みました。

各説明文章

1. の「ホーム」は、「家、家庭、故郷」。「自宅」や「住む場所」を意味しています。

2. の「ホームは、「本拠地」。「ホームグラウンド」や「本拠地」を指しています。チームが所属する場所や施設のことです。

3. の「ホーム」は、「老人ホームなどの略」。「老人ホーム」のような福祉施設を意味しています。

4. の「ホーム」は、「プラットホームの略」。「鉄道駅のプラットホーム」を指しています。

※テストデータは ChatGPT で作成した意味区分のものと同様

ChatGPT によって作成させた意味区分と Wiktionary の意味区分で異なる点として、区分の細かさが挙げられる。例えば、セットでは、ChatGPT で作成させた意味区分は、4 つの意味だけであり、どれも特定の場面のみで使われるようなものはないが、Wiktionary の意味区分では 8 つの意味があり、2. スカッシュ・テニス・卓球・バレーボールなどの試合の一区切りや、8. 猟犬が伏せの動作により獲物を指示することなど、ある特定の場面でのみ使われるような意味も含まれている。このように ChatGPT で作成した意味区分よりも、人間の定義した意味区分のほうが、より汎化されていない、局所的な意味を含む区分となっている。

以上の意味区分で実験を行い、生成 AI で作成した意味区分と、実際に人間が定義した意味区分で、各 LLM を用いた意味分類の有効性に違いがあるのかを確認することを目的とする。

3.4 実験結果

3.4.1 実験 1

以下の表 1 から表 6 に実験 1 の結果を示す。表 1 から表 6 では [”ポイント”,”カット”,”ホーム”,”セット”,”モデル”,”アップ”] の 6 つの単語を対象とした Few-shot Learning の結果を示している。実験 1 では各プロンプトに役割を与えず、実験 2 で役割を与えた際の実験結果との

比較対象とした。

表 1　各 LLM での”ポイント”における選択肢の出力頻度

正解：2	1	2	3	4	5
gpt-3.5-turbo	15	24	23	20	18
gpt-4o-mini	31	4	16	47	2
gpt-4o	60	40	0	0	0
Gemini-Pro	87	3	2	3	5
Swallow	28	25	9	9	29

表 2　各 LLM での”カット”における選択肢の出力頻度

正解：3	1	2	3	4
gpt-3.5-turbo	10	57	28	5
gpt-4o-mini	69	0	31	0
gpt-4o	0	0	100	0
Gemini-Pro	88	0	12	0
Swallow	30	19	34	17

表 3　各 LLM での”ホーム”における選択肢の出力頻度

正解：2	1	2	3	4
gpt-3.5-turbo	2	51	34	13
gpt-4o-mini	1	31	56	12
gpt-4o	1	99	0	0
Gemini-Pro	23	26	29	22
Swallow	27	30	17	26

表 4　各 LLM での”セット”における選択肢の出力頻度

正解：4	1	2	3	4
gpt-3.5-turbo	11	65	10	14
gpt-4o-mini	30	4	2	64
gpt-4o	12	0	0	88
Gemini-Pro	11	8	29	52
Swallow	28	23	26	23

表 5　各 LLM での”モデル”における選択肢の出力頻度

正解：2	1	2	3	4
gpt-3.5-turbo	8	27	44	21
gpt-4o-mini	2	35	20	43
gpt-4o	0	100	0	0
Gemini-Pro	8	86	5	1
Swallow	22	27	26	25

表 6　各 LLM での”アップ”における選択肢の出力頻度

正解：4	1	2	3	4	5
gpt-3.5-turbo	3	6	38	31	22
gpt-4o-mini	4	14	3	75	0
gpt-4o	0	0	0	100	0
Gemini-Pro	6	12	79	2	1
Swallow	20	23	13	16	28

3.4.2　実験 2

以下の表 7 から表 12 に実験 2 の結果を示す。表 7 から表 12 では [”ポイント”,”カット”,”ホーム”,”セット”,”モデル”,”アップ”] の 6 つの単語を対象とした言語分析家という役割を与えた際の Few-shot Learning の結果を示している。実験 2 では各プロンプトに”言語分析家”という役割を持たせる記述を組み込み、これによって回答の精度が向上することを期待した。表中において数字が赤色で色付けされている部分は、実験 1 に比べて正解の選択肢の出力頻度が上がったことを示しており、数字が青色で色付けされている部分は、逆に出力頻度が下がったことを示している。なお表 13 では、実験 1、実験 2 における LLM 毎の平均正解率を示している。

表 7　各 LLM での”ポイント”における選択肢の出力頻度

正解：2	1	2	3	4	5
gpt-3.5-turbo	3	15	37	38	7
gpt-4o-mini	29	4	7	60	0
gpt-4o	3	97	0	0	0
Gemini-Pro	68	7	5	3	16
Swallow	15	34	7	19	25

表 8　各 LLM での”カット”における選択肢の出力頻度

正解：3	1	2	3	4
gpt-3.5-turbo	2	72	19	7
gpt-4o-mini	61	0	39	0
gpt-4o	0	0	100	0
Gemini-Pro	55	0	44	1
Swallow	35	20	31	14

表 9　各 LLM での”ホーム”における選択肢の出力頻度

正解：2	1	2	3	4
gpt-3.5-turbo	2	54	34	10
gpt-4o-mini	0	1	29	70
gpt-4o	0	100	0	0
Gemini-Pro	30	32	25	13
Swallow	25	25	21	29

表 10　各 LLM での”セット”における選択肢の出力頻度

正解：4	1	2	3	4
gpt-3.5-turbo	4	64	11	21
gpt-4o-mini	27	14	19	40
gpt-4o	0	0	0	100
Gemini-Pro	59	9	13	19
Swallow	17	23	30	30

表 11　各 LLM での”モデル”における選択肢の出力頻度

正解：2	1	2	3	4
gpt-3.5-turbo	10	19	48	23
gpt-4o-mini	1	41	5	53
gpt-4o	0	100	0	0
Gemini-Pro	2	92	6	0
Swallow	24	19	19	38

表 12　各 LLM での”アップ”における選択肢の出力頻度

正解：4	1	2	3	4	5
gpt-3.5-turbo	6	12	24	34	24
gpt-4o-mini	1	3	1	95	0
gpt-4o	0	0	0	100	0
Gemini-Pro	12	6	43	23	16
Swallow	6	12	79	2	1

表 13　各 LLM での平均正解率

平均正解率	実験 1	実験 2
gpt-3.5-turbo	28.50	27.00
gpt-4o-mini	40.00	36.67
gpt-4o	87.83	99.50
Gemini-Pro	30.17	36.17
Swallow	25.83	23.50

以上より実験 1 と実験 2 の結果を比較すると、表 13 より gpt-4o はすべてのプロンプトにおいて正解率が向上し、次いで Gemini-Pro が”セット”という単語に関するプロンプト以外で正解率の向上が見られた。その他の LLM では、単語によって正解率が上下し、さらに平均正解率も低下しているため、役割を与えることが有意に正解率の向上に寄与したとは断言できない。gpt-3.5-turbo-0125 と Swallow は全体的に精度が低く、最も正解率が高い場合でも gpt-3.5-turbo-0125 の”ホーム”に関するプロンプトで 54% であった。

注目するべき点として、”ホーム”に関するプロンプトでは、gpt-4o-mini の正解率が役割を与えることで 31% から 1% まで低下しており、さらに”セット”に関するプロンプトでは、Gemini-Pro の正解率が 52% から 19% まで低下していることが挙げられる。Swallow に関しても、”アップ”に関するプロンプトで、正解率が 16% から 2% まで低下している。対照的に”アップ”に関するプロンプトでは、gpt-4o-mini の正解率が 75% から 95% まで上昇し、Gemini-Pro の正解率が 2% から 23% まで上昇した。”カット”に関するプロンプトでも、Gemini-Pro の正解率が 12% から 44% まで上昇している。

3.4.3　実験 3

実験 3 として、Wiktionary で実際に定義されている意味区分に基づいて、Few-shot Learning を行った際の各 LLM における意味分類の有効性を検証するため、ChatGPT で作成した文章を学習データとして Few-shot Learning を行った。

まず、Wiktionary 内で [”ポイント”,”カット”,”ホーム”,”セット”,”モデル”,”アップ”] の持つ意味群の中から、専門性の高い意味 (汎用的に用いられていない意味) や、英単語の意味として使われている意味を除いたものを選択し、各意味を持つ単語を含む文章を ChatGPT に作成させた。例としてポイントという単語は、”点。ドット”,”場所。地点”,”要点”,”小数点”,”(鉄

道) 転轍器、転路器”,”(単位) 活字の大きさの単位”,”点。得点”,”尖頭器。一端または両端がとがっている石器や骨角器のこと”,”(アイスホッケー) 敵のゴールの両側で攻撃を行うポジション”,”(ラクロス) ゴールキーパーの前方で敵のシュートを防ぐポジション”という 10 個の意味が Wiktionary 内で定義されているが、専門性の高い意味を持つ”(鉄道) 転轍器、転路器”,”(単位) 活字の大きさの単位”,”(アイスホッケー) 敵のゴールの両側で攻撃を行うポジション”,”(ラクロス) ゴールキーパーの前方で敵のシュートを防ぐポジション”と、英単語の point の意味を持つ”小数点”は除外した。

具体的には下記のようなプロンプトで必要な文章を作成している。

プロンプトの例

文脈中の意味が「ドット」、「場所。地点」、「要点」、「得点」、「尖頭器」であるポイントという単語を含む例文をそれぞれの意味に対応して作成してください。

以下が上記のプロンプトに対する回答である。

回答の一部

・ドット (点): プリンターの解像度は、1 インチあたりのポイント数で表されます。
・場所・地点: 次のポイントでバスを降りてください。そこが目的地です。
・要点: 彼のプレゼンテーションの主要なポイントをメモしてください。
・得点: チームは試合の最後に決定的なポイントを獲得し、勝利しました。
・尖頭器（武器の先端や尖った部分）: その古代の槍のポイントは非常に鋭く、慎重に取り扱う必要があります。

次に以上の方法で作成した文章から、各文章中の対象単語の意味を説明する文章を以下のようなプロンプトによって ChatGPT によって作成させた。

プロンプトの例

・プリンターの解像度は、1 インチあたりのポイント数で表されます。
・次のポイントでバスを降りてください。そこが目的地です。
・彼のプレゼンテーションの主要なポイントをメモしてください。
・チームは試合の最後に決定的なポイントを獲得し、勝利しました。
・その古代の槍のポイントは非常に鋭く、慎重に取り扱う必要があります。
以上の文章中のポイントの意味を説明してください。

以下が上記のプロンプトに対する回答である。

回答の一部

1. プリンターの解像度は、1 インチあたりのポイント数で表されます。
・ドット (点): ここでは「ポイント」はプリンターの解像度を示すために使用される「ドット（点)」を指しています。
2. 次のポイントでバスを降りてください。そこが目的地です。
・場所・地点: この場合の「ポイント」は地理的な「場所」や「地点」を意味しています。
3. 彼のプレゼンテーションの主要なポイントをメモしてください。
・要点: ここでは「ポイント」は「主要な点」や「要点」を意味しています。4. チームは試合の最後に決定的なポイントを獲得し、勝利しました。
・得点: この文では「ポイント」はスポーツや競技における「得点」を指しています。
5. その古代の槍のポイントは非常に鋭く、慎重に取り扱う必要があります。
・尖頭器（武器の先端や尖った部分）: ここでの「ポイント」は「尖った部分」や「先端」を意味しています。

以上で作成されたデータを基に、図 1 から図 3 に示す各 LLM のテンプレートの形に添って整形し、実験 3 で扱うプロンプトとした。

以下の表 14 から表 19 に実験 3 の結果を示す。実験 1、実験 2 では BCCWJ から抽出した文章における対象単語の意味区分を ChatGPT に作成させ、各 LLM の意味分類に対する有効性検証を行ったのに対し、実験 3 では Wiktionary で人間が定義した意味区分における各 LLM の意味分類に対する有効性検証を行った。テストデータは実験 1、実験 2 と同様のものとしている。なお表 20 では、生成 AI が作成した意味区分と、実際に人間が定義した意味区分における意味分類の精度を比較するため、実験 1 から実験 3 の LLM 毎の平均正解率を示している。

表 14　各 LLM での”ポイント”における選択肢の出力頻度

正解：4	1	2	3	4	5
gpt-3.5-turbo	5	16	30	30	19
gpt-4o-mini	42	0	54	4	0
gpt-4o	33	1	60	5	1
Gemini-Pro	33	12	45	5	5
Swallow	26	11	12	27	24

表 15　各 LLM での”カット”における選択肢の出力頻度

正解：1	1	2	3	4	5
gpt-3.5-turbo	6	20	6	50	18
gpt-4o-mini	99	0	1	0	0
gpt-4o	99	1	0	0	0
Gemini-Pro	85	4	3	5	3
Swallow	21	19	13	23	24

表 16　各 LLM での”ホーム”における選択肢の出力頻度

正解：4	1	2	3	4
gpt-3.5-turbo	0	62	14	24
gpt-4o-mini	0	0	0	100
gpt-4o	0	0	0	100
Gemini-Pro	94	3	3	0
Swallow	17	23	17	43

表 17　各 LLM での”カット”における選択肢の出力頻度

正解：3	1	2	3	4	5	6	7	8
gpt-3.5-turbo	0	13	4	1	4	7	7	64
gpt-4o-mini	0	0	100	0	0	0	0	0
gpt-4o	0	0	100	0	0	0	0	0
Gemini-Pro	1	0	84	2	5	3	4	1
Swallow	19	9	6	13	5	8	29	11

表 18　各 LLM での”モデル”における選択肢の出力頻度

正解：1	1	2	3	4
gpt-3.5-turbo	1	18	45	36
gpt-4o-mini	1	0	99	0
gpt-4o	0	0	90	10
Gemini-Pro	72	3	20	5
Swallow	18	19	30	33

表 19　各 LLM での”アップ”における選択肢の出力頻度

正解：1	1	2	3	4	5	6
gpt-3.5-turbo	15	27	3	25	28	2
gpt-4o-mini	75	10	1	13	1	0
gpt-4o	99	1	0	0	0	0
Gemini-Pro	100	0	0	0	0	0
Swallow	27	15	16	9	13	20

表 20　各 LLM での平均正解率

平均正解率	実験 1	実験 2	実験 3
gpt-3.5-turbo	28.50	27.00	13.33
gpt-4o-mini	40.00	36.67	63.17
gpt-4o	87.83	99.50	67.17
Gemini-Pro	30.17	36.17	57.67
Swallow	25.83	23.50	23.67

以上の実験 3 の結果から、主に OpenAI API の LLM と Gemini-Pro に関して、意味分類の精度が各対象単語で大きく異なることが確認でき、LLM によって得意、または苦手な単語があるのではないかという推察が生まれた。例えば”ホーム”という単語に関して、gpt-4o-mini 及び gpt-4o の正解率は 100% であるのに対し、Gemini-Pro の正解率は 0 であり、誤選択肢 1 の出力が 94% である。また、”モデル”という単語では、Gemini-Pro の正解率は 72% であるのに対し、gpt-4o-mini、gpt-4o の正解率は 0 であり 90% 以上が誤選択肢 3 を出力している。このように単語によって LLM 間の正解率が大きく異なり、間違った選択肢を 90% 以上出力するようなハルシネーションが見られる。

さらに、gpt-4o-mini と Gemini-Pro において、生成 AI で作成した各単語の意味区分よりも実際の人間が作成した意味区分を学習に用いるほうが、平均正解率が高いという結果になった。どちらも 20% 以上向上しており、カタカナ語に関しては、学習データ全体に対する生成 AI が作成したデータの割合を低くすると、これらの LLM の生成精度が向上すると期待できる。

ここで実験 3 を行った後に気づいたこととして、ポイントの意味区分に正しいといえる選択肢がなかったことが挙げられる。実験 1,2 でのポイントの意味区分は、「金融や経済の専門用語」、「商品やサービスにおける得点」、「重要な要素」、「競技やゲームにおける得点」、「場所」であるのに対し、実験 3 で利用した意味区分では、「ドット (点)」、「場所・地点」、「要点」、「得点」、「尖頭器 (武器の先端や尖った部分)」である。そしてテストデータは、”ココの優待は株を長期保存するとポイントがＵＰするんだけど、その前に倒産〜って事にならないと良いけどね〜。”というものであり、期待した回答の意味としては「商品やサービスにおける得点」である。しかし、実験 3 では、期待する回答に最も近い「得点」という意味の説明文章を、”スポーツや競技における獲得できる「得点」を指しています。”と記述してしまっているため、正し

いといえる選択肢がなく、期待した正解の選択率が低くなってしまったと考えられる。このため、”選択肢以外の回答を行う場合は出力は 0 としてください。”とプロンプト内に記述し、与えられた選択肢以外を選べるようにした。以下の表 21 に各 LLM での”ポイント”における選択肢の出力頻度を示す。

表 21　各 LLM での”ポイント”における 0 を含む選択肢の出力頻度

正解：0	0	1	2	3	4	5
gpt-3.5-turbo	2	8	30	39	10	11
gpt-4o-mini	0	41	0	53	6	0
gpt-4o	87	2	3	7	1	0
Gemini-Pro	5	28	4	55	7	1
Swallow	11	25	20	7	19	18

表 21 を見ると、gpt-4o の出力の約 90% が 0 であり期待した出力になったが、ほかの LLM では、0 の出力が見られるものの、全体の約 10% 以下の出力しか見られなかった。このことから以上の方法は gpt-4o では間違った選択肢を選ぶハルシネーションの解消に役立つと思われるが、他の LLM では難しいと考えられる。

4. まとめ

今回の研究実験により、生成 AI が作成した意味区分を用いた意味分類では、gpt-4o が圧倒的に正解率が高く、他の LLM では正解率が低いという結果になった。これは意味区分を作成した生成 AI に gpt-4o を起用したことが原因である可能性があるため、今後は Gemini などの生成 AI で作成した意味区分での Few-shot Learning による意味分類を行いたいと思う。また gpt-4o、Gemini-Pro においては役割を与えることによる精度の向上が見られたため、プロンプトを改良することでさらに精度の向上を期待できると思われる。そして実験 3 で行った Wikitionary による人間が定義した意味区分を利用した意味分類では、gpt-4o の正解率が最も高かったものの、gpt-4o-mini と Gemini-Pro では生成 AI の作成した意味区分よりも平均正解率が大幅に向上したことが確認できたため、今後の研究では人間が定義した意味区分を利用した場合を主に取り扱いたい。

今回扱った Swallow は Llama2 に日本語データを追加学習させた、いわゆる日本語に特化した LLM であるため、カタカナ語の意味分類では精度の高さを期待していたが、実際は正解率があまり高くなかったことが検証の結果確認できた。そのため他の日本語に特化した LLM などでも同様の実験を行い、共通点や相違点を見つけ、プロンプトで改善できる点や苦手なカタカナ語などを調査したいと思う。

文 献

[1] Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Ya-

suharu Den. Balanced corpus of contemporary written japanese. *Language resources and evaluation*, Vol. 48, pp. 345–371, 2014.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, June 2019.

[3] Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. A robustly optimized BERT pre-training approach with post-training. In Sheng Li, Maosong Sun, Yang Liu, Hua Wu, Kang Liu, Wanxiang Che, Shizhu He, and Gaoqi Rao, editors, *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pp. 1218–1227. Chinese Information Processing Society of China, August 2021.

[4] Marco Maru, Simone Conia, Michele Bevilacqua, and Roberto Navigli. Nibbling at the hard core of Word Sense Disambiguation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4724–4737, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[5] Terra Blevins and Luke Zettlemoyer. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1006–1017, Online, July 2020. Association for Computational Linguistics.

[6] Sakae Mizuki and Naoaki Okazaki. Semantic specialization for knowledge-based word sense disambiguation. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3457–3470, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[7] Ming Wang and Yinglin Wang. A synset relation-enhanced framework with a try-again mechanism for word sense disambiguation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6229–6240, Online, November 2020. Association for Computational Linguistics.

[8] Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szyd, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, Anna Kocoń, BartKoptyra, Wiktoria Mieleszczenko-Kowszewicz, Piotr Mi, Marcin Oleksy, Maciej Piasecki, Radliński, Konrad Wojtasik, StanisWoźniak, and PrzemysKazienko. Chatgpt: Jack of all trades, master of none. *Information Fusion*, Vol. 99,

p. 101861, 2023.

[9] Haoqiang Kang, Terra Blevins, and Luke Zettlemoyer. Translate to disambiguate: Zero-shot multilingual word sense disambiguation with pretrained language models. In Yvette Graham and Matthew Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1562–1575, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

関連 URL

LLM『CyberAgentLM3』`https://huggingface.co/cyberagent/calm3-22b-chat`
LLM『Youko』`https://huggingface.co/rinna/llama-3-youko-8b`
クラウドサービス『Modal』`https://modal.com/`
多機能辞典『Wiktionary』`https://ja.wiktionary.org/wiki`