日本語学習者辞書語釈の自動生成

井手 佑翼[1,2] Adam Nohejl[2,1] Joshua Tanner[3] 谷中 瞳[2,4,5] Christopher Lindsay[6] 渡辺 太郎[1]
[1]NAIST [2]理化学研究所 [3]Resolve Research [4]東京大学 [5]東北大学 [6]Serpenti Sei Japan
ide.yusuke.ja6@is.naist.jp

概要

学習者辞書は、見出し語の意味を平易な単語のみで記述した辞書であり、言語学習等に有用な資源である。しかし、日本語については、高品質な学習者辞書はいまだ制作されていない。そこで、本研究は、日本語学習者辞書の自動編纂に向けて語釈の生成に取り組む。我々は、Few-shot プロンプティングと反復的平易化を組み合わせた、シンプルな語釈生成手法を提案する。実験結果から、提案手法は、辞書語釈としての高い品質と語彙的平易性を同時に実現できることを示す。

1 はじめに

辞書の語釈は、単語の意味を理解するため、あるいは語義曖昧性解消などの分野の研究のために重要な資源である。しかし、これらの資源はしばしば数千を超える語釈を必要とするため、これを人手で作成する作業は膨大な労力を要する。この課題を踏まえ、複数の先行研究が語釈の自動生成に取り組んできた。たとえば、百科事典の解説文 [1] や俗語辞典の語釈 [2] の生成に関する研究が存在する。

これに対し、我々は、学習者辞書語釈の生成（learner's dictionary definition generation, LDDG）、中でも日本語の LDDG に取り組む。学習者辞書は、非母語話者向けに語釈を平易な単語のみで構成した辞書である。英語辞書（英英辞書）の中では主要なカテゴリの一つであり、主要な英語辞書出版社 6 社はみな学習者辞書を発行している [3]。また、学習者辞書は言語学習、特に言語産出能力を向上させるために有効だとされている [4]。学習者辞書用の平易な語釈は、言語学習で広く使われる単語帳の中でも活用できると期待される。同時に、日本語については、海外の日本語学習者数が 2024 年に過去最多の 400 万人を記録する [5] など、近年、大きく学習者が増加している。以上の背景にも関わらず、高品質な日本語学習者辞書はいまだ制作・提供されていない。

表 1 日本語 LDDG の入出力例。見出し語、読み、品詞、語釈の出所は、すべて D3J [9]。

入力	出力
満たす, みたす, verb	“（人の希望などを）満足させる。 （条件などを）達成する。”, “（入れ物などを）いっぱいにする。”

本研究では、Few-shot プロンプティングと反復的平易化手法である IterSim を組み合わせた LDDG 手法を提案する。IterSim は、[6] に基づくシンプルな手法で、大規模言語モデル（LLM）を用いて、語釈の品質を保ったまま語釈中の難解な単語を一つずつ取り除く。実験結果から、提案手法により、辞書語釈としての高い品質と語彙的平易性を同時に実現できることを示す。語釈の品質の点では、人間の編集者により書かれた Wiktionary を上回る品質が実現されていることを示唆する。

2 関連研究

辞書編纂学の分野では、ChatGPT の登場が、辞書編纂の自動化に対する関心を高めた [7]。たとえば [8] が、ChatGPT を用いて語釈および例文からなる辞書項目を生成する手法について調査した。しかし、管見の限り、語彙制約付きの語釈生成としての学習者辞書語釈生成に取り組んだ研究は存在しない。

3 学習者辞書語釈生成の定式化

我々は、LDDG を、語彙制約付きの辞書語釈生成（dictionary definition generation, DDG）として定式化する。DDG は、見出し語、品詞、読みの三つ組を入力として、語釈の配列を出力するタスクである。ただし、日本語 DDG において、見出し語がひらがなとカタカナのみで構成される場合、読みは追加的な情報をもたらさないので、本研究ではこれを省略する。表 1 に、日本語 LDDG の入出力の例を示す。

ここでの語彙制約とは、与えられた定義語彙（defining vocabulary）と呼ばれる単語のリストに含

表 2　語釈生成のためのプロンプトの抜粋。{guidelines} プレースホルダには、詳細なガイドラインが挿入される。

```
Describe the definitions of the senses of the given headword
using simple words for learners of Japanese. Include only
senses that are expected to be familiar to most native Japanese
speakers today. [...]

# Guidelines
{guidelines}
```

アルゴリズム 1　IterSim

Input: Headword h, definition def
Output: Simplified definition $currDef$

$compWords \leftarrow \textsc{FindComplexWords}(def)$
$currDef \leftarrow def$
for all $w \in compWords$ **do**
  $isSuccess, simDef \leftarrow \textsc{Simplify}(currDef, w, h)$
  if $isSuccess$ **then**
    $currDef \leftarrow simDef$
  end if
end for
return $currDef$

まれない単語は使用できないという制約である。本研究では、定義語彙として TUBE16K [9] を用いる。TUBE16K は、日本語コーパス中の出現頻度が高い 16,000 語および日本語の語彙や文法の解説に有用な 10 語からなる。この 16,000 語は、日本語学習者辞書データセットである D3J [9] の参照語釈を人手で作成する際に、最小限必要な語彙サイズとして設定されたものである。また、語彙サイズを 16,000 語程度に制限することにより、語釈中の難解な単語を同じ辞書の別の項目で説明することも容易になる。本研究では、TUBE16K に含まれる単語を平易、そうでない単語を難解とみなし、平易な単語のみを使った語釈の生成に取り組む。

4　LLM を用いた語釈生成

本研究では、Few-shot プロンプティングと反復的平易化を組み合わせた手法を提案する。

4.1　シングルターン・プロンプティング

本研究では、第一に、シングルターンのプロンプティング、つまり Zero-shot プロンプティングおよび Few-shot プロンプティングの性能を検証する。LLM に入力するプロンプトの抜粋を表 2 に示す。Few-shot プロンプティングでは、D3J のデモデータ（demonstration set）からランダムに入力（見出し語、読み、品詞）と出力（語釈）のペアを 5 件抽出し、事例としてプロンプトに含める。

アルゴリズム 2　単語 w を対象とした平易化

Input: Definition def, complex word w, headword h
Output: Success flag $isSuccess$, simplified definition $simDef$

$inferenceCount \leftarrow 0$
$cWords \leftarrow \textsc{FindComplexWords}(def)$
$cCount \leftarrow \textsc{CountComplexWords}(def)$
$bannedWords \leftarrow []$
while $inferenceCount \leq 2$ **do**
  $simDef \leftarrow \textsc{Inference}(h, def, w, bannedWords)$
  if $\textsc{CountComplexWords}(simDef) \leq cCount$ **then**
    return $True, simDef$
  end if
  $cWords2 \leftarrow \textsc{FindComplexWords}(simDef)$
  $newCompWords \leftarrow \textsc{Set}(cWords2) - \textsc{Set}(cWords)$
  if $\textsc{Length}(newCompWords) > 0$ **then**
    $bannedWords \leftarrow bannedWords + newCompWords$
  else
    $def \leftarrow simDef$
  end if
  $inferenceCount \leftarrow inferenceCount + 1$
end while
return $False, \varepsilon$

表 3　平易化のためのプロンプト。波括弧はプレースホルダを表す。

```
The given definition contains a complex word that could be
difficult for learners. Rewrite it without using the complex
word, ensuring that (1) the revised version remains an accurate
and fluent representation of the headword's sense and (2) the
revised version does not contain the banned words if any are
provided. [...]

Headword: {headword}
Definition: {definition}
Target word: {target_word}
Banned words: {banned_words}
Simplified Definition:
```

4.2　反復的平易化

第二に、IterSim によって、シングルターン・プロンプティングで出力した語釈の平易性を高めることができるかどうかを検証する。IterSim は、[6] に基づく手法で、LLM を用いて、語釈の品質を保ちながら、難解な単語すなわち定義語彙に含まれない単語を一つずつ取り除く。シングルターン・プロンプティングでは単に易しい単語を用いるよう指示していただけであったのに対し、IterSim では取り除くべき単語が何であるか明示する。[6] との違いとして、本研究では、平易化の過程で別の難解な単語が現れるケースへの対処も行う。

アルゴリズム 1 に、IterSim のアルゴリズムの概要を示す。IterSim は、まず入力された初期語釈に対し

て単語分割と TUBE16K に含まれない単語の抽出を行い、得られた単語を難解語として記録する（1 行目）。続いて、語釈の正確性を維持しながら、難解語を一つずつ取り除く試行を行う（2–8 行目）。

4 行目の SIMPLIFY() は、シングルターン・プロンプティングの際と同じ LLM および表 3 のプロンプトを用いながら、Zero-shot プロンプティングを最大で 2 回行う。アルゴリズム 2 に、この関数のアルゴリズムを示す。まずプロンプティングによる推論を行う（6 行目）。この推論によって難解語の数が減少した場合、*isSuccess* の値として *True* を、*simDef* の値として推論の出力を返す（8 行目）。難解語の数が減少しなかった場合、推論によって語釈中に新しい難解語が現れたかどうかを調べる。（1）新しい難解語が現れていた場合、これをプロンプト（表 3）の中の banned_words に追加したうえで（13 行目）再度推論を行う。（2）新しい難解語が現れていなかった場合、LLM が期待していたのとは異なる単語を平易化してしまった等の理由で、もともとの平易化対象だった *w* が出力中に残っていたと判断できる。そこで、*def* を推論の出力で上書きしたうえで（15 行目）再度推論を行う。以上の手続きを 2 回繰り返しても難解語が減少しなかった場合、*isSuccess* の値として *False* を返す（18 行目）。

5 実験設定

5.1 推論

実験に使用する LLM は、モデル選定の時点で Nejumi Leaderboard 4[1] において上位に入っていた、プロプライエタリモデル 2 つ、すなわち GPT-5.1 および Claude Sonnet 4.5（以降 Claude）と、オープンウェイトモデル 2 つ、すなわち Qwen3-32B [10]（以降 Qwen）および Llama-3.3-Swallow-70B-Instruct-v0.4 [11] とする。

推論時には、原則として、各モデルのデフォルトの Reasoning 設定を使用する。Qwen については、Reasoning をオンに設定すると出力を JSON に制限することが難しくなるためオフに設定する。温度は、すべてのモデルで 0.0 とする。オープンウェイトモデルを使用する際は、bitsandbytes[2] ライブラリを用いて 4 ビット量子化を行う。

1） https://wandb.ai/llm-leaderboard/nejumi-leaderboard4/reports/Nejumi-LLM-4--VmlldzoxMzc1OTk1MA

2） https://github.com/TimDettmers/bitsandbytes

表 4 シングルターン・プロンプティングの結果（各観点の平均スコア）。Few-shot のスコアは、異なる事例を用いた 3 回の推論の平均。± は標準偏差を表す。Wiktionary のガイドライン遵守性スコアは、Wiktionary の編集者は我々のガイドラインを参照しながら語釈を執筆したわけではないため、無効としている。

	総合	真実性	網羅性	具体性	遵守性
Zero-shot					
Claude	87.2	89.5	94.6	73.4	91.2
GPT	86.1	93.5	97.6	57.1	96.3
Llama	65.9	68.9	56.0	66.9	71.9
Qwen	57.9	62.4	74.2	42.7	52.3
Few-shot					
Claude	90.8±1.0	91.5±0.8	91.9±1.1	88.0±1.4	92.0±1.5
GPT	87.8±0.6	93.2±0.5	96.7±0.9	65.5±2.6	95.6±0.8
Llama	63.8±3.4	60.8±4.2	51.9±1.0	72.2±3.0	70.5±5.3
Qwen	60.7±8.9	61.8±9.4	67.0±10.2	49.0±6.9	65.2±9.1
Wikt	—	69.7	90.8	76.3	—

5.2 評価

生成された語釈は、[9] の評価フレームワークを用いて評価する。このフレームワークは、真実性（truthfulness）、網羅性（coverage）、語義具体性（sense specificity）、ガイドライン遵守性（guideline compliance）の 4 観点で、LLM-as-a-judge を用いて語釈の評価を行う。データは、250 件の単語と語釈のペアを含む D3J のテストデータを用いる。LLM-as-a-judge モデルとしては、GPT-5.1 を用いる。また、比較対象として、寛容なライセンスのもとで公開された人手の語釈であり、D3J にバンドルされている Wiktionary の語釈も評価する。

一方、平易性の尺度として、TUBE16K 比率を定義する。TUBE16K 比率は、3 節で議論した定義語彙 TUBE16K に含まれる単語のみで構成された語釈の比率とする。

6 実験結果

6.1 シングルターン・プロンプティング

表 4 に、シングルターン・プロンプティングの結果を示す。Few-shot プロンプティングは、Llama を除くすべてのモデルにおいて、Zero-shot プロンプティングを超える総合スコア（4 観点のスコアの平均）を記録した。プロプライエタリモデルは、オープンウェイトモデルを 20–30%上回った。最も高性能な組み合わせは、Claude を用いた Few-shot プロンプティング（Claude Few-shot）であった。この組

表 5　Claude を用いた Few-shot プロンプティングの出力例。

見出し語	参照語釈	出力語釈	注
揺らぐ	“揺れる。”, “(物事や気持ちなどが）不安定になる。”	“(物が）左右や前後に動く。揺れる。”, “(気持ちや考えなどが）不安定になる。”	すべての観点で満点。
苦痛	“苦しみ。苦しくて嫌なこと。”	“心や体の苦しみや痛み。”, “我慢するのが難しいほど嫌なこと。”	語義具体性に課題。
鼻	“顔の中央にある、呼吸をしたり匂いをかいだりするための器官。”	“顔の中央にある、呼吸をしたり匂いを感じたりするための器官。”, “物の先端の部分。”	真実性に課題。

表 6　IterSim の結果。使用したモデルは Claude。± は標準偏差を表す。

	総合	真実性	網羅性	語義具体性	ガイドライン遵守性	TUBE16K
Few-shot	90.8±1.0	91.5±0.8	91.9±1.1	88.0±1.4	92.0±1.5	90.3±0.9
Few-shot + IterSim16K	90.9±1.0	91.3±1.0	91.6±1.6	87.9±1.0	92.7±1.6	99.8±0.2

表 7　Claude を用いた Few-shot プロンプティングおよび IterSim の出力例。太字は難解語を表す。

見出し語	参照語釈	出力語釈（IterSim 後 ← IterSim 前）
先祖	“家系や血統などの、初期または前の世代の人々。”	“自分より前の世代で、血のつながりがある人。...” ← “自分より前の世代の**血縁**者。...”

み合わせは、GPT を用いた Few-shot プロンプティングと比べると、網羅性が 20%程度高かった。さらに、Wiktionary を 3 つの観点で上回り、この手法による語釈が Wiktionary 編集者の語釈より高品質であることを示唆する結果となった。表 5 に、Claude Few-shot の出力例を示す。「揺らぐ」に対する語釈が適切に生成されていることが分かる。

一方、課題も残されている。表 4 から、いずれの手法についても、4 観点の中では語義具体性のスコアが低いことが読み取れる。表 5 に、語義具体性が低い出力の例として「苦痛」の出力を示す。二つの語釈はともに精神的な苦しみを表しており、過剰に意味が重複していると判断できる。また、辞書の利用者に誤った知識を伝えてはいけないという点を踏まえると、真実性は非常に重要な観点であるが、Claude Few-shot の真実性スコアは 91.5%に留まった。真実性が低い出力の例として「鼻」の出力を示す。二つ目の語釈「物の先端の部分」は、一般に使われる語義ではなく、参照語釈にも含まれない。このような語釈が生成された理由としては、車の（ボンネット等を含む）先端部分が英語で nose と呼ばれており、その訳語である「鼻」に対して、上記のような語釈が出力された可能性が考えられる。しかし、学習者が優先順位の高い重要な語義に集中できるようにするという観点では、このような語釈は学習者辞書に含めるべきではない。以上のような点を解消することが、今後の課題と言える。

また、Claude Few-shot の TUBE16K 比率は平均で 90.3%であり、一部の語釈中に難解な単語が含まれていることが示された。この比率を IterSim で改善することができるかどうかを次節で分析する。

6.2　反復的平易化

表 6 に、Claude Few-shot の出力と、それをさらに IterSim で平易化した語釈のスコアの比較を示す。この結果から、IterSim により、4 つの観点のスコアは維持したまま TUBE16K 比率をほぼ 100%に引き上げることができたことが分かる。すなわち、語釈の品質を保ったまま平易性を高めることができた。表 7 に、Claude Few-shot および IterSim の出力の例を示す。難解語である「血縁」が適切に平易化されていることが分かる。

7　おわりに

本研究では、LLM による Few-shot プロンプティングと反復的平易化手法 IterSim を組み合わせた、学習者辞書語釈の生成手法を提案した。実験により、提案手法は、与えられた語彙制約をほぼ完全に満たしながら高品質な語釈を生成できることを示した。

本研究は、自動学習者辞書編纂の実現を長期的な目標とし、その第一歩として語釈生成に取り組んだが、これと並んで重要なタスクとして、辞書向けの例文生成がある [12]。学習者辞書においては例文も平易であることが望ましいが、IterSim はこの例文を平易化するための手法としても活用の余地があると考えられる。

謝辞

本研究は、JST 次世代研究者挑戦的研究プログラム JPMJSP2140 の支援を受けたものである。研究の過程で、国立国語研究所の山崎誠氏に重要なコメントをいただいた。

参考文献

[1] Shonosuke Ishiwatari, Hiroaki Hayashi, Naoki Yoshinaga, Graham Neubig, Shoetsu Sato, Masashi Toyoda, and Masaru Kitsuregawa. Learning to describe unknown phrases with local and global contexts. 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. 3467–3476, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[2] Ke Ni and William Yang Wang. Learning to explain non-standard English words and phrases. In Greg Kondrak and Taro Watanabe, editors, **Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)**, pp. 413–417, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

[3] Reinhard Heuberger. Learners' dictionaries: History and development; current issues. In **The Oxford Handbook of Lexicography**. Oxford University Press, 11 2015.

[4] Michael Rundell. Dictionary use in production. **International journal of lexicography**, Vol. 12, No. 1, pp. 35–53, 1999.

[5] 国際交流基金. 2024 年度「海外日本語教育機関調査」結果, 2025.

[6] Masashi Oshika, Makoto Morishita, Tsutomu Hirao, Ryohei Sasano, and Koichi Takeda. Simplifying translations for children: Iterative simplification considering age of acquisition with LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Findings of the Association for Computational Linguistics: ACL 2024**, pp. 8567–8577, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[7] Gilles-Maurice de Schryver. Generative ai and lexicography: The current state of the art using chatgpt. **International Journal of Lexicography**, Vol. 36, No. 4, pp. 355–387, 10 2023.

[8] Robert Lew. Chatgpt as a cobuild lexicographer. **Humanities and Social Sciences Communications**, Vol. 10, No. 1, pp. 1–10, 2023.

[9] Yusuke Ide, Adam Nohejl, Joshua Tanner, Hitomi Yanaka, Christopher Lindsay, and Taro Watanabe. Towards automated lexicography: Generating and evaluating definitions for learner's dictionaries. **arXiv preprint arXiv:2601.01842**, 2026.

[10] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. **arXiv preprint arXiv:2505.09388**, 2025.

[11] Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. Continual pre-training for cross-lingual llm adaptation: Enhancing japanese language capabilities. In **Proceedings of the First Conference on Language Modeling**, COLM, University of Pennsylvania, USA, October 2024.

[12] Bill Cai, Ng Clarence, Daniel Liang, and Shelvia Hotama. Low-cost generation and evaluation of dictionary example sentences. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, pp. 3538–3549, Mexico City, Mexico, June 2024. Association for Computational Linguistics.