Exploring the Potential of Prompt-Based Method for Kanji-Kana Conversion in Japanese Braille Translation

Micah Kitsunai[1] Deborah Watty[1] Shu-Kai Hsieh[1]
[1]Graduate Institute of Linguistics, National Taiwan University, Taipei, Taiwan
{r11142010, r11142012, shukaihsieh}@ntu.edu.tw

Abstract

This study focuses on the challenge of polyphone disambiguation in the automatic translation process of Japanese into Japanese Braille and explores the applicability of Large Language Models (LLM) to this task. We propose a method that includes additional information obtained by employing a morphological analysis tool when prompting the LLM. The results indicate that this combined approach enhances accuracy beyond the predictive capabilities of morphological analysis tools alone and also surpasses the standalone use of the LLM.

1 Introduction

Even though text-to-speech applications are very widespread, Braille is still an important communication tool for people with visual impairments. With every new publication, there is a need for Braille translation, and this demand continues to grow.

Braille functions with 6 dots forming a single cell, allowing for 64 different combinations based on whether each dot is raised or not. Since the number of possible characters is limited, more than one Braille cell is sometimes used to represent a single print symbol [1, 2]. A Japanese Braille character typically corresponds one-to-one with a single Kana character[1)] (see Table 1). Therefore, it is necessary to convert Kanji into Kana for Braille translation. While there is Braille Kanji (漢点字) that uses 8 dots, it is not widely used and does not require Kana conversion, so it is not considered in this study.

Japanese is commonly written in a combination of Kanji and Kana. Kanji characters have multiple readings (examples are shown in Table 2), known as On'yomi (Chinese-derived reading) and Kun'yomi (original Japanese reading). Additionally, there are polyphones, which are words with the same spelling but different readings [3].When the wrong reading is chosen for a word during Braille conversion, it is up to the reader to identify this mistake, slowing down the reading process.

Despite the development and utilization of numerous automatic Braille translation tools, their accuracy is not perfect, necessitating manual corrections. The primary objective of this study is to propose a method for applying Large Language Models (henceforth referred to as LLMs) to automatic Braille translation that does not require fine-tuning and can be implemented at a low cost.

Table 1 Example of Kana Conversion

今日		は	晴れ	
today		TOPIC	sunny	
キョ (kyo)	ウ (u)	ハ (ha)	ハ (ha)	レ (re)
キョ (kyo)	ウ (u)	ワ (wa)	ハ (ha)	レ (re)
⠈⠪	⠉	⠄	⠥	⠛

1.1 Literature Review

A key challenge in Braille translation is accurately predicting the appropriate readings of Kanji. Rule-based methods, such as Liblouis [4], exist but require the continuous addition of new rules. Alternative approaches involve using Neural Machine Translation Technology [5, 6] or BERT for polyphone disambiguation [7]. There are also studies using pre-trained language models for polyphone disambiguation in text-to-speech [8], but they require extensive Braille data and training. Additionally, research using corpora with Kana annotations [6] and their validations [9] have been conducted.

With the advent of large language models, prompt engineering is an emerging approach to a variety of problems [10, 11, 12]. For the purpose of Braille translation, Reference [13] used a prompt-based method for the disam-

[1)] Kana refers to Hiragana or Katakana, which are phonograms.

Table 2 Classification of Multiple Readings in Japanese

Word	Reading	Type	Note
日	hi	On vs. Kun	On'yomi
日	nichi	On vs. Kun	Kun'yomi
会社	**ka**isha	voiced vs. unvoiced	Single use
会社	**ga**isha	voiced vs. unvoiced	Compound word (e.g. 株式会社)
辛い	karai	polyphone	(Meaning) spicy
辛い	tsurai	polyphone	(Meaning) sadness
昨日	kinou	synonym	Soft expression
昨日	sakujitsu	synonym	Formal expression

biguation of polyphone characters in Taiwanese Mandarin. The study employed syntactic analysis results and external dictionary data as part of the prompt given to an LLM, recording a higher accuracy rate than generic rule-based methods.

2 Proposed Method

In this paper, we adapt the method used in [13] to the use case of Japanese Braille translation. Figure 1 illustrates the pipeline of the proposed method. In the following section each step in the pipeline is explained in more detail and Table 3 summarizes the various problems that need to be solved for Braille conversion and which part of the pipeline addresses each problem.

2.1 Morphological Analysis

Initially, texts containing a mix of Kanji and Kana are analyzed using the open-source morphological analysis engine Mecab [14], employing its standard IPA dictionary. This analysis yields a range of linguistic information for each morpheme, including Part of Speech (POS), reading, and phonetic notation (phonetics). The process of converting to kana encompasses two distinct types: "Kana as written form" and "Kana as spoken form." In Braille translation, the latter principle is applied and the sounds are transcribed exactly as heard. As illustrated in Table 2, a characteristic of this method is the representation of "は" (ha) as "わ" (wa). Mecab's phonetics information, readily usable and also employed in other studies [5], forms the basis of our system. Differences in On'yomi and Kun'yomi readings, as shown in Table 2, are often correctly predicted through this analysis. However, since the determination of readings for homographs is not perfect, subsequent methods are needed to enhance the accuracy.

2.2 Dictionary Entries

Dictionary entries for words are obtained using the JishoAPI [15], an API that provides results from Jisho,[2] an online Japanese-English dictionary. Since most Kana and some Kanji readings are unique, it is not necessary to add this information to all words. To identify words with unclear readings, we focus on words containing Kanji after morphological analysis, extracting only those with multiple readings or multiple entries in the dictionary. The acquired dictionary data can be seen as an example in the prompt illustrated in Figure 3.

2.3 Prompt Design

In this proposed method, the library Langchain [16] was used, which facilitates application development utilizing Large Language Models. For words with unclear readings, the meanings and possible readings obtained from JishoAPI were inserted into the prompt, as shown in Figure 3. This prompt is processed by the LLM to predict the correct readings of the Kanji.

2.4 Conversion to Braille

The readings obtained from the prompt and from the dictionary entries are then combined. Since Kana and Braille characters are almost exclusively associated as one-to-one pairs, the conversion is straightforward from here on out. The results are displayed as shown in Figure 3.[3]

3 Evaluation and Results

For verification, we used the polyphones included in the test data as keys and counted whether their readings were correctly predicted when using each method.

3.1 Test Data

We constructed a dataset of 67 short sentences, each containing one polyphone (from a list of 18 different polyphone characters), from the Balanced Corpus of Contemporary Written Japanese [17]. To ensure that the context around the target word could be referenced, 20 words before and after each polyphone were extracted. The target words were selected based on [9].

[2] https://jisho.org/
[3] Our code demo site are available at https://github.com/muoegu/jp2braille

Table 3 Strategies in the Proposed Method of This Study

Problem	Tool Used	Solution
Insufficient Kana Reading Data for LLM	Dictionary Data	Adding Dictionary Data to Prompt
Excessive Use of API	Mecab	Obtaining Basic Readings
Acquisition of Phonetic Readings	Mecab	The analysis results include Phonetic Readings, which conform to the rules of braille
LLM Token Limits	Dictionary Data	Narrowing Down to Words with Ambiguous Readings

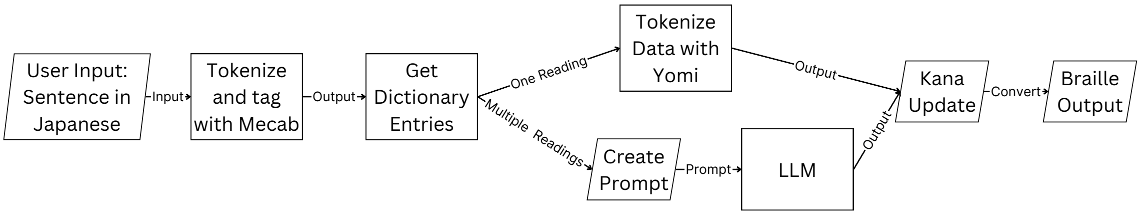
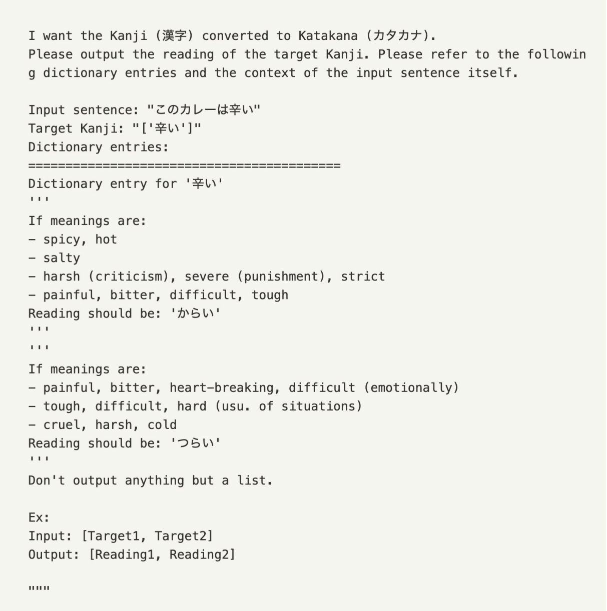


Figure 1 Pipeline of the Proposed Method



```
I want the Kanji (漢字) converted to Katakana (カタカナ).
Please output the reading of the target Kanji. Please refer to the followin
g dictionary entries and the context of the input sentence itself.

Input sentence: "このカレーは辛い"
Target Kanji: "['辛い']"
Dictionary entries:
==========================================
Dictionary entry for '辛い'
'''
If meanings are:
- spicy, hot
- salty
- harsh (criticism), severe (punishment), strict
- painful, bitter, difficult, tough
Reading should be: 'からい'
'''
'''
If meanings are:
- painful, bitter, heart-breaking, difficult (emotionally)
- tough, difficult, hard (usu. of situations)
- cruel, harsh, cold
Reading should be: 'つらい'
'''
Don't output anything but a list.

Ex:
Input: [Target1, Target2]
Output: [Reading1, Reading2]

"""
```

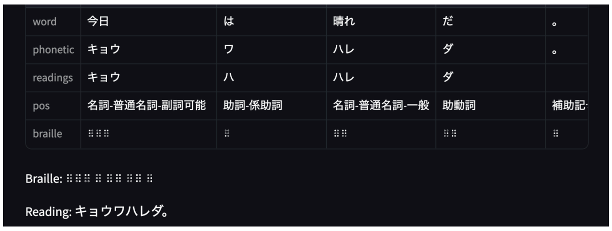
Figure 2 An Example of a Prompt

3.2 Evaluation Method

Table 5 presents the results by morphological analysis alone, GPT-3.5 and GPT-4 alone, as well as GPT-3.5 and GPT-4 paired with morphological analysis.

3.2.1 Mecab

Mecab allows for the prediction of readings for each morpheme, so its usability as a morphological analysis tool was tested. Instances were observed where the context-based judgment was insufficient. For some characters, the tool selected the same reading for all example sentences. Additionally, errors were noted where, despite being technically correct, uncommon readings were chosen.



word	今日	は	晴れ	だ	。
phonetic	キョウ	ワ	ハレ	ダ	。
readings	キョウ	ハ	ハレ	ダ	
pos	名詞-普通名詞-副詞可能	助詞-係助詞	名詞-普通名詞-一般	助動詞	補助記
braille	⠈⠪⠉	⠄	⠥⠛	⠐⠕	⠲

Braille: ⠈⠪⠉ ⠄ ⠥⠛ ⠐⠕ ⠲

Reading: キョウワハレダ。

Figure 3 An Example of the Results

3.2.2 OpenAI Models

Using OpenAI's API, attempts were made to obtain Kanji readings using GPT-3.5 and the latest GPT-4 models. We adopted a zero-shot-prompting method, providing prompts that simply instruct the text to be converted to Katakana. The OpenAI models used were `gpt-3.5-turbo` and `gpt-4`. With both the GPT-3.5 and GPT-4 models, there were instances of seemingly random combinations of On'yomi and Kun'yomi, or entirely random, non-existent readings.

3.2.3 Mecab + OpenAI Models

Finally, we combined the morphological analysis with Mecab with prompting the LLM as described in the previous section. We tested the same models as in our zero-shot approach. In the experiments conducted, the GPT models used were all set with a Temperature of 0 for the purpose of reproducibility.

The accuracy improved with this method. Specifically, referencing Mecab and dictionaries helped prevent the generation of non-existent readings by GPT.

4 Discussion

The low accuracy rate of the method using only prompting with GPT-3.5 is likely due to a shortage of data on

Table 4 Improvements gained from the Proposed Method

Word (Reading)	Mecab	LLM	Mecab + LLM	diff
図 zu	zu	*tou	zu	Referencing Mecab prevents hallucinations by the LLM
私 watashi	*watakushi	watashi	watashi	The LLM corrects the choice of an uncommon reading by Mecab

Table 5 Comparison of Scores and Accuracy for Each Model

Model	Score	Accuracy
Mecab	41/67	61.19%
GPT-3.5	29/67	43.28%
GPT-4	48/67	71.64%
Mecab ＋ GPT-3.5	43/67	64.17%
Mecab ＋ GPT-4	50/67	74.63%

pairs of Kanji characters and their readings in such language models. However, the significant improvement in results with GPT-4 suggests that the ability to predict readings using only prompts greatly depends on the quality and quantity of the data.

In the methodology employed in this study, the risk of generating non-existent words is low due to the use of reading information and dictionary data based on Mecab's morphological analysis, as demonstrated in Table 4. Furthermore, it was observed that more accurate reading predictions become possible by performing context-based corrections using LLMs after the selection of readings by Mecab. This approach effectively combines the strengths of both systems: Mecab's precise morphological analysis capabilities and the complex contextual understanding of LLMs. Thus, it compensates for the shortcomings of each system and offers a more balanced approach.

However, it was observed that the accuracy rate for synonyms with different readings is still low. This could be due to the semantic categorization used not adequately addressing these specific cases. Initially, it was thought that the system had flexible response capabilities, but the effect on synonyms turned out to be limited. Nonetheless, in Braille conversion, since there is little change in meaning, it is unlikely to be a significant burden for users.

5 Conclusion

In this study, we explored the potential of using LLMs for Kanji-to-Kana conversion in the context of automatic Japanese Braille translation. Compared to the standalone use of morphological analysis tools like Mecab and language models such as GPT, the method proposed in this research — incorporating dictionary search results for polyphones into prompts and then inputting these into an LLM — demonstrated higher accuracy. The improvement in the accuracy of automatic Kana assignment for Kanji, as explored in this study, is expected to contribute not only to Braille translation but also to assist children and Japanese language learners who are not yet proficient in Kanji, as well as to enhance text-to-speech accuracy.

Table 6 Examples and Classification of Errors

Word (Reading)	Error	Category	Method
人気 hitoke	*ninnki	Unable to judge the context	Mecab
私 watashi	*watakushi	uncommon pronunciation	Mecab
人気 hitoke	*jinnke	non-existent reading	LLM
人事 jinnji	*ninnji	non-existent reading	LLM
昨日 kinou	*sakujitsu	synonym	all

In the future, there is potential in expanding the dictionary database and incorporating comprehensive textual context and word usage examples into the prompts. The current methodology is confined to single-sentence analysis due to LLM token limits, but an essential objective moving forward would be to extend this approach to full-text analysis and improvements in translation speed while maintaining the accuracy of analysis.

References

[1] 日本点字委員会. 日本点字表記法2018年版. 社, 2018.

[2] 特定非営利活動法人全国視覚障害者情報提供施設協会. 点訳のてびき 第4版. 社, 2019.

[3] 西山浩気, 山本和英, 中嶋秀治. 読み曖昧性解消のためのデータセット構築手法. 人工知能学会全国大会論文集 第32回 (2018), pp. 4Pin152–4Pin152. 一般社団法人 人工知能学会, 2018.

[4] Christian Egli. Liblouis–a universal solution for braille transcription services. In **Proceedings of Daisy 2009 Conference**, 2009.

[5] Yuko Shimomura, Hiroyuki Kawabe, Hidetaka Nambo, and Shuichi Seto. **Braille Translation System Us-**

ing Neural Machine Translation Technology I - Code Conversion, p. 335345. Springer International Publishing, June 2019.

[6] Hiroyuki Kawabe, Yuko Shimomura, and Shuichi Seto. Braille translation system using neural machine translation technology II – code conversion of kana-kanji mixed sentences. In **Proceedings of the Fifteenth International Conference on Management Science and Engineering Management: Volume 1 15**, pp. 417–426. Springer, 2021.

[7] 佐藤文一, 喜連川優ほか. 事前学習済み bert の単語埋め込みベクトルによる同形異音語の読み誤りの改善. 研究報告アクセシビリティ (AAC), Vol. 2020, pp. 1–5, 2020.

[8] Rem Hida, Masaki Hamada, Chie Kamada, Emiru Tsunoo, Toshiyuki Sekiya, and Toshiyuki Kumakura. Polyphone disambiguation and accent prediction using pre-trained language models in japanese tts front-end, 2022.

[9] 喜連川優佐藤文一. 大規模振り仮名注釈付きコーパスを用いた同形異音語の読み分類. 言語処理学会第28回年次大会 (NLP2022), pp. 1878–1883, 2022.

[10] Bertalan Meskó. Prompt engineering as an important emerging skill for medical professionals: tutorial. **Journal of Medical Internet Research**, Vol. 25, p. e50638, 2023.

[11] Vivian Liu and Lydia B Chilton. Design guidelines for prompt engineering text-to-image generative models. In **Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems**, pp. 1–23, 2022.

[12] Unggi Lee, Haewon Jung, Younghoon Jeon, Younghoon Sohn, Wonhee Hwang, Jewoong Moon, and Hyeoncheol Kim. Few-shot is enough: exploring chatgpt prompt engineering method for automatic question generation in english education. **Education and Information Technologies**, pp. 1–33, 2023.

[13] Deborah Watty, Micah Kitsunai, and Shu-Kai Hsieh. Prompt-based translation of chinese into taiwanese mandarin braille. In **2023 International Conference on Asian Language Processing (IALP)**, pp. 56–61. IEEE, 2023.

[14] MeCab MeCab. Yet another part-of-speech and morphological analyzer, 2006.

[15] jisho-api. https://github.com/pedroallenrevez/jisho-api/, 2004. Accessed: 2023-01-12.

[16] Harrison Chase. Langchain. https://github.com/langchain-ai/langchain, 2022. Accessed: 2023-01-12.

[17] 前川喜久雄. Kotonoha 『現代日本語書き言葉均衡コーパス』 の開発 (¡ 特集¿ 資料研究の現在). 日本語の研究, Vol. 4, No. 1, pp. 82–95, 2008.