Prompt-Based Translation of Chinese into Taiwanese Mandarin Braille

Deborah Watty
Graduate Institute of Linguistics
National Taiwan University
Taipei, Taiwan
r11142012@ntu.edu.tw

Micah Kitsunai
Graduate Institute of Linguistics
National Taiwan University
Taipei, Taiwan
r11142010@ntu.edu.tw

Shu-Kai Hsieh
Graduate Institute of Linguistics
National Taiwan University
Taipei, Taiwan
shukaihsieh@ntu.edu.tw

***Abstract*—In automated Braille translation, accommodating linguistic nuances and the rules peculiar to Braille across various languages poses considerable challenges. Mandarin Chinese stands out in this aspect due to its necessity to ascertain the appropriate pronunciation of characters based on context. Although rule-based algorithms have historically dominated this space, recent empirical evidence highlights the efficacy of statistical approaches and the emergent exploration of Large Language Model (LLM)-based techniques. This paper explores the potential advantages of leveraging a prompt-based strategy for the automated translation from Mandarin Chinese to Taiwanese Mandarin Braille. As a methodology, we devised a script capable of ingesting a Chinese sentence and subsequently generating a prompt that comprises the Zhuyin of unequivocal characters and dictionary definitions for those with polysemous readings. Utilizing a set of 103 test sentences, we assessed the precision with which GPT-3.5, GPT-4, and Liblouis (a widely-recognized open-source rule-based Braille translator) ascribed readings to polyphonic characters. Our findings revealed that, notwithstanding certain inconsistencies in the GPT-3.5 outputs, the extended GPT-4 model exhibited superior performance compared to Liblouis.**

***Index Terms*—Braille, Large Language Models (LLMs), Taiwanese Mandarin, polyphone disambiguation**

I. Introduction

Braille stands as the predominant tactile writing system employed by the visually impaired community. Transcribing printed text into Braille is intricate, mainly due to the absence of a direct one-to-one mapping between each Unicode character and a Braille counterpart. Moreover, the governing rules for this conversion exhibit variation across languages and regions.

First, a single print character might necessitate multiple Braille characters for accurate representation. A Braille character is composed of six dots, which can either be in a raised or flat configuration, leading to a total of 64 distinguishable characters. This arrangement falls short when attempting to encompass the myriad of print characters, including lower and upper case letters, punctuation marks, numerals, and specialized symbols. Additionally, typographical nuances, such as italicization used for emphasis, demand the deployment of supplementary Braille characters. Consequently, a frequent one-to-many relationship emerges between print and Braille characters.

Conversely, a multitude of Braille systems incorporates various abbreviations to optimize space, considering the expansive nature of Braille text and its requirement for robust paper or its exhibition on single-line displays. Such practices often culminate in a many-to-one association between print and Braille characters.

The intricacies encountered during the automation of such conversions are inherently diverse, with the degree of challenge contingent upon the characteristics of the writing system in tandem with its Braille counterpart.

In the case of Chinese, a stark distinction emerges between its print and Braille forms. While printed Chinese relies on ideographic symbols, Chinese Braille adopts a phonetic notation. A salient challenge in the Chinese script is the existence of numerous characters with polyphonic tendencies, where accurate pronunciation hinges heavily on contextual surroundings. This inherent ambiguity introduces complexities in automating the transition from Chinese print to Braille.

Given the existing technological constraints of currently available tools, a rigorous review by a seasoned Braille specialist is imperative before disseminating a converted document. As expounded in the findings of [1], a critical examination of the allocation and administration of resources for the visually impaired within libraries in Taiwan reveals salient deficiencies. The research indicates that not only are the existing resources for this demographic subpar, but they are also unequally distributed among a select number of institutions. One of the notable obstacles impeding the seamless integration of resources is technical challenges.

Most automated Braille translation software are predominantly anchored on rule-based algorithms, with the open-source library Liblouis being a frequently adopted choice [2]. Though certain research endeavors have explored machine learning methodologies, utilizing corpora for Mainland Chinese Braille [3], [4], there appears to be a dearth of similar undertakings for Taiwanese Mandarin Braille. While recent advancements have witnessed the deployment of Large Language Models (LLMs) for character disambiguation, their application has not yet been extended to the domain of Taiwanese Mandarin Braille conversion [5]–[7].

The goal of our paper is to explore the efficacy of prompt-based techniques in facilitating the translation of Chinese text into Taiwanese Mandarin Braille, placing a specific emphasis on the disambiguation of polyphonic characters.

Subsequently, we provide a succinct elucidation of the most important rules governing Taiwanese Mandarin Braille. This is followed by an overview of the functionalities of Liblouis, complemented by a synopsis of contemporary research focused on both statistical and LLM-oriented methodologies. Concluding the discussion, we delineate the methodology behind our prompt generation, elucidate the evaluation metrics adopted to assess our findings, and deliberate on potential implementations of our proposed framework. [1]

II. Taiwanese Mandarin Braille

Taiwanese Mandarin Braille is based on Zhuyin, the most commonly used phonetic spelling of Mandarin in Taiwan. Given the Zhuyin transcription of a Chinese character, it can easily be converted into its Braille form. The basic conversion rules are as follows:

- Each Zhuyin initial is mapped to a Braille character. Three pairs of Zhuyin initials share a corresponding Braille character (ㄍ/ㄐ, ㄘ/ㄑ, ㄙ/ㄒ) since their pronunciation is easily disambiguated by looking at the final of the syllable.
- Medials and finals are combined and mapped to a single Braille character. If there is no final (e.g. for the character "知" whose Zhuyin is ㄓ), a placeholder character is used.
- Tones are always written at the end of the syllable, including the first tone, which is omitted in Zhuyin.
- With exceptions for foreign words and special characters, there are no spaces between words.

The full set of rules including those for punctuation, special characters and foreign words can be found in [8]. Note that Taiwanese Mandarin Braille differs considerably from the Braille version used in Mainland China, which includes spaces between words and only marks tones for certain syllables.

III. Literature Review

Pioneering efforts in the domain of automatic Braille translation can be traced back to the 1960s [9], [10]. Subsequent to these initial forays, the field has witnessed the development of a myriad of both proprietary and open-source Braille transcription software. Historically, these applications have predominantly hinged on rule-based methodologies. However, recent years have witnessed a paradigm shift with researchers gravitating towards the deployment of machine learning models specifically tailored for Chinese Braille translation.

A. Rule-based approach

To the best of our knowledge, Liblouis stands out as a prominent open-source, rule-based library, boasting support for a multitude of languages. It forms the bedrock for several accessibility tools, ranging from converters integral to Braille displays to dedicated Braille translation applications [11]. The predominant hurdle tethered to rule-based methods pertains to the complexities of maintaining and updating an extensive rule set. This challenge is starkly manifested in the sheer volume of the translation table for Taiwanese Mandarin Braille within Liblouis, which encompasses tens of thousands of lines [12].

Such translation tables play a pivotal role in guiding the transcription process across varied languages. They comprise Braille translations corresponding to distinct print character sequences, potentially accompanied by annotations or keywords. These annotations are instrumental in specifying the sequence in which the stipulated rules are executed.

As an example, the character "覺" can either be read as ㄐㄧㄠˋ (Pinyin: *jiào*, e.g. in the word "睡覺", to sleep) or ㄐㄩㄝˊ (*jué*, e.g. in the word "感覺", feeling). The table contains an entry for the word "睡覺" in the form

```
#x7761   #x89BA   24-1246-5   13-246-5
(  睡       覺         ⠊⠫⠐        ⠅⠪⠐   )
```

where the Chinese characters are indicated by their Unicode codes and the Braille characters are indicated by lists of numbers representing the raised dots, separated by dashes.

The term "睡覺" can indeed be dissected into separate entities, exemplified in expressions like "睡一覺" (to take a nap). While there exists a dedicated entry for this specific phrase, it is inherently infeasible to encapsulate every conceivable insertion between these characters within the translation table. For instance, the phrase "睡個好覺" (to take a good nap) remains absent from the table, resulting in Liblouis attributing an erroneous pronunciation to it[2]. Consequently, any endeavor to rectify a mistranslation mandates augmenting or refining the existing rule set, thereby perpetuating the expansion of the translation tables.

Conversely, one of Liblouis' distinct advantages lies in its rapid translation capability, a feature attributed to its core architecture being directly crafted in the C programming language.

B. Machine learning-based approach

The landscape of automatic Braille translation has been punctuated by the deployment of both statistical methodologies and neural network architectures. Inherently, strategies within this paradigm demand substantial datasets in Braille for effective execution. The acquisition of these datasets, contingent on the language in question, can sometimes present challenges due to their rarity or specificity.

For instance, Zhang et al. [3] leveraged a corpus comprising Chinese words paired with their corresponding Braille translations. They employed an n-gram based strategy to facilitate the translation process, specifically targeting Mainland Chinese Braille.

Wang et al. [4] advanced an innovative Deep Neural Network (DNN) framework designed for an end-to-end translation process targeting Mainland Chinese Braille. Interestingly, in a user-centric study assessing the efficacy of their model, participants rated the quality of the automated translations in close parity with that of manually crafted reference translations.

[1] Our code and test set are available at https://github.com/deborahwatty/gpt_taiwan_braille

[2] as observed in Liblouis version 3.26.0

It is noted, however, that the aforementioned methodologies all focus on Mainland Chinese Braille. The domain of employing corpus-driven strategies for the automated translation into Taiwanese Mandarin Braille remains notably under-researched and represents a frontier yet to be extensively charted.

C. LLM-based approach

In recent years, the allure of utilizing pre-trained language models, exemplified by architectures such as BERT [13], for polyphone disambiguation tasks has been embraced by multiple research groups.

Chen et al. [5] innovatively deploy contextually-sensitive representations of polyphonic characters, as derived from BERT, to serve as input for a subsequent neural network. This model is then tasked with predicting the appropriate pronunciation of the focal character.

In a somewhat divergent approach, Zhang et al. [6] refine a pre-trained Chinese BERT model by incorporating novel characters. These characters are strategically designed to depict the diverse pronunciations inherent to polyphonic characters, enabling the model to accurately disambiguate.

Venturing into the realm of Mainland Chinese Braille translation, a seminal study by Yu et al. [7] deserves mention. The team adopted pre-trained Transformer and GPT architectures, rigorously evaluating their performance against varied sizes of fine-tuning datasets. This comprehensive exploration underscores the potential of these models in the context of Braille translation.

IV. Prompt Generation

Since dictionaries and text databases generally do not contain Braille but do have Zhuyin, we make use of the fact that Zhuyin is easily converted into Braille, as described in Section II, and use only Zhuyin in our prompt and the output of the LLM. To convert the output string from Zhuyin to Braille, we use a rule-based Python script.

Given an input sentence in Traditional Chinese, we generate a single prompt that asks the LLM to convert the sentence to Zhuyin. The prompt is made up of several parts:

1) Zhuyin for characters for which the correct reading can be identified without using an LLM. This can be the case either because the word is not a polyphone (e.g. “萌”, which only has the reading ㄇㄥˊ / méng), or because the character is part of a compound with a single correct reading (e.g. the polyphone “地” in the compound “地震”, which only has the reading ㄉㄧˋㄓㄣˋ / dì zhèn). To find these readings, we tokenize and tag the input sentence using spaCy [14] and then make API calls to MoeDict (萌典) [15] to find the reading. MoeDict is specifically tailored to feature Taiwanese Mandarin pronunciations, thereby rendering it our favored resource.
2) Dictionary entries are taken from MoeDict for characters or compounds with multiple possible readings.
3) Instructions for the formatting of the output. The purpose of this is to ensure that the Zhuyin output can be converted by our Braille conversion script. We use the LangChain output parser [16] and include a one-shot example for the output format in the field description.

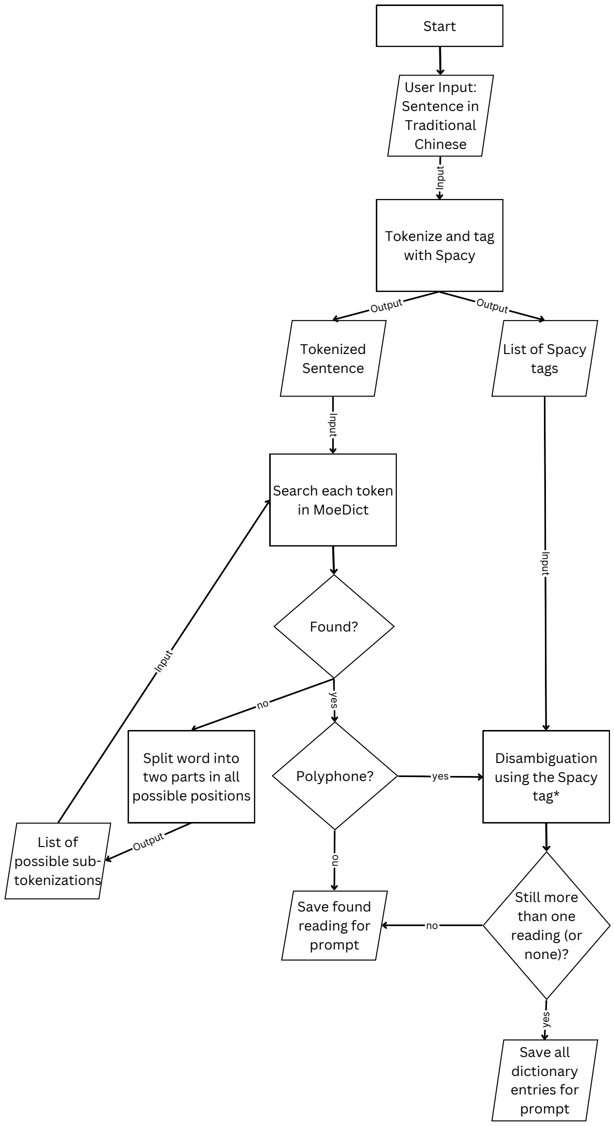

Fig. 1. Visualization of the collection of the information needed to create the prompt.

*Disambiguation using the spaCy tag is only possible if the token comes directly from the spaCy tokenization and has not been split. Otherwise, this step is skipped.

The process of generating parts 1 and 2 is visualized in Fig. 1. An example of a resulting prompt is shown in Fig. 2. We predict that given the information included in the prompt, high-performance LLMs such as GPT-3.5 and GPT-4 will be able to leverage the meaning of a character given its wider context to identify the correct reading for polyphone characters with a high degree of accuracy.

~~~
I want to find the Zhuyin for the following sentence:
很舒服的一覺
I have already tokenized the sentence and found the following pronunciations:
很-ㄏㄣˇ
舒服-ㄕㄨ ˙ㄈㄨ
的-˙ㄉㄜ
一-ㄧ
覺-unknown
The Zhuyin for the characters marked as unknown is ambiguous.
The character 覺 can have the following meanings:

1. Verb meaning "睡醒。". In this case, the Zhuyin is "ㄐㄩㄝˊ".
2. Verb meaning "醒悟、感悟。". For example: 覺悟, 後知後覺. In this case, the Zhuyin is "ㄐㄩㄝˊ".
3. Verb meaning "知曉、感受到、意識到。". For example: 發覺, 警覺, 自覺, 不知不覺. In this case, the Zhuyin is "ㄐㄩㄝˊ".
4. Verb meaning "啟發、告訴。". In this case, the Zhuyin is "ㄐㄩㄝˊ".
5. Noun meaning "感官受刺激後對事物的辨識能力。". For example: 知覺, 味覺, 幻覺, 錯覺. In this case, the Zhuyin is "ㄐㄩㄝˊ".
6. Noun meaning "賢智之人。". For example: 先覺. In this case, the Zhuyin is "ㄐㄩㄝˊ".
7. Noun meaning "睡眠。". For example: 睡午覺, 睡大覺. In this case, the Zhuyin is "ㄐㄧㄠˋ".
8. Noun meaning "量詞。計算入睡次數的單位。". For example: 睡了一覺. In this case, the Zhuyin is "ㄐㄧㄠˋ".

Given this information, can you convert the sentence to Zhuyin, including the ambiguous ones? The output should be a markdown code snippet formatted in the following schema, including the leading and trailing "```json" and "```":

```json
{
   "zhuyin": string  // The sentence in Zhuyin only: replace every character by its Zhuyin, keeping the order of the original sentence, e.g. 少了幾件 -> ㄕㄠˇ ˙ㄌㄜ ㄐㄧˇ ㄐㄧㄢˋ
}```
~~~

Fig. 2. Example of a prompt generated for the input phrase “很舒服的一覺”. The highlighted parts change depending on the input phrase:

- Red: The input phrase.
- Green: Known Zhuyin after tokenization with spaCy and lookup of the tokens in MoeDict.
- Yellow: This phrase is omitted if there are no characters with ambiguous readings.
- Blue: Information about characters with multiple readings taken from MoeDict.

V. Evaluation and Results

For our evaluation dataset, we chose the Chinese Word Net (CWN 2.0) database [17] since the Zhuyin of the target character is included with every example sentence. We generated the test set by choosing 25 common polyphone characters and randomly selecting sentences with 20 or fewer characters from their example sentences in the CWN database. Our final test set contains a total of 103 sentences for 21 unique polyphone characters.

We translated all sentences into Taiwanese Mandarin Braille using Liblouis 3.26.0 as a baseline. The target polyphone characters were assigned the correct reading in 81.6% of the sentences. We also generated the prompts as described above for each sentence and tested them with both GPT-3.5 (text-davinci-003) and GPT-4. For the sake of consistency across multiple identical queries, we set the temperature to 0 in both cases.

Even though we used short sentences for our tests, 35 of the 103 sentences produced prompts that exceeded the token limit of text-davinci-003 (4097 tokens). Of the remaining 68 sentences, the correct reading was assigned to the target character in 86.8% of the sentences. We did not run into issues with the token limit using GPT-4, which assigned the correct reading to 87.4% of the target characters. The results are summarized in Fig. 3.

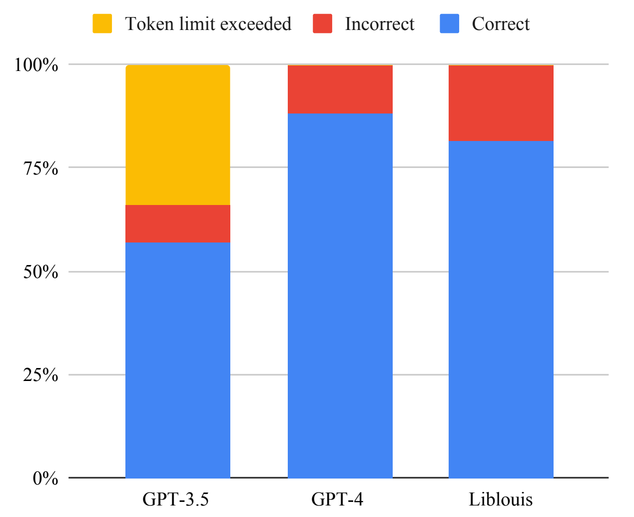

Fig. 3. Percentage of test sentences in which the correct reading was found for the target polyphone character (N=103).

The output of the GPT-3.5 model was inconsistent. Out of the cases where the token limit was not reached, the number of tokens in the translation did not match the number of tokens in the original string in 26.5% of cases, mainly due to characters being skipped or reduplicated in the translation process. In other instances, the reason for the mismatch in the number of tokens was inconsistencies in the formatting of the translated string, such as omission of spaces. In one instance, the returned Zhuyin string was interspersed with the original Chinese characters. These kinds of outputs introduce errors into the subsequent translation into Braille.

GPT-4 did not have these issues. The two cases in which the input and output length did not match were related to punctuation (two cases of a missing period at the end of a sentence and one case of missing spaces next to parentheses). The GPT-4 model was also able to handle an instance of an unexpected Zhuyin form in the prompt where MoeDict had returned a string containing two variants for the word “什麼” (“ㄕㄣˊ ˙ㄇㄜ （又音）ㄕㄜˊ ˙ㄇㄜ”). While GPT-3.5 simply copied the full entry into the resulting string, GPT-4 automatically chose only the first option (“ㄕㄣˊ ˙ㄇㄜ”).

VI. Discussion

We set out to investigate whether prompt engineering could be useful for Braille translation, focusing on the disambiguation of polyphone Chinese characters. The results indicate that given enough information in a prompt, the ability of both GPT-3.5 and GPT-4 to disambiguate polyphone characters is comparable to or even slightly better than that of Liblouis. It is notable that this level of accuracy was achieved with a fairly simple prompt, while Liblouis is an ongoing project that has been improved upon by the community for years and already has tens of thousands of lines in the rule base for Taiwanese Mandarin. We predict that with some improvements to our prompting method and improvements to LLMs over the course of the next few years, LLM-based solutions could outperform traditional table-based methods for Chinese Braille translation.

A. GPT-3.5 vs. GPT-4

A major problem we encountered with the GPT-3.5 model was that the output format was inconsistent in multiple cases, including characters being skipped or reduplicated, such as in the following sentence:

Sentence	...販	售	所	得	將	全	部...
GPT-3.5	...ㄈㄢˋ	ㄕㄡˋ	ㄙㄨㄛˇ	ㄉㄜˊ	**omitted**	ㄑㄩㄢˊ	ㄅㄨˋ...
GPT-4	...ㄈㄢˋ	ㄕㄡˋ	ㄙㄨㄛˇ	ㄉㄜˊ	ㄐㄧㄤ	ㄑㄩㄢˊ	ㄅㄨˋ...

This type of error is arguably even more disruptive to the reading flow than an incorrect reading of a character known to the reader. For this reason, we opted not to rerun the tests with a model based on GPT-3.5 that has a higher token limit.

We did not encounter this issue with the GPT-4 model. To the contrary, as described above, in one instance GPT-4 was even able to compensate for an error that occurred upstream in the pipeline when it chose one of two readings that were given for a character that had been deemed unambiguous during preprocessing.

B. Limitations

From our experimentation with the GPT-3.5 model, it becomes manifest that the token constraint for a prompt is as a significant impediment in our methodology. The limit of 4097 tokens is reached very quickly when we need multiple dictionary entries within a single prompt. Although our endeavors with the GPT-4 model (which has higher token limit) did not elicit any discernible issues, sentences longer than those in our test set might instigate challenges analogous to those encountered with the GPT-3.5 model.

Consequently, it is imperative to devise a mechanism to fragment sentences into more concise segments, ensuring that they remain within the token threshold. This segmentation is paramount for the consistent and dependable translation of texts with extended sentences, but it is no trivial task; as explained in Section III-A, the essential cues needed for determining the accurate pronunciation of a character can often be situated at a considerable distance from the character in focus. Addressing this nuanced challenge is a pivotal objective we are keen to explore in forthcoming research endeavors.

A salient limitation inherent in our methodology is the inability to rectify a specific erroneous output. This is in stark contrast with tools like Liblouis, where a single error can often be fixed by adding a single rule to the table. Due to the opaque nature of LLMs, this is not possible with our approach.

Additionally, the latency introduced by the multiple API invocations for each sentence - which can extend to several seconds - considerably undermines the practicality of our method for real-time translation applications. This is especially pertinent in scenarios requiring instantaneous translations, such as utilizing a Braille display to interpret a text not previously rendered in Braille. Unless there is a notable acceleration in the responsiveness of these API calls or an alternative that makes them redundant altogether, more rapid solutions such as Liblouis are poised to remain the preferred option for such immediate tasks.

C. Future Work

Our long-term vision is to evolve our system to a stage where comprehensive texts, replete with foreign words, unique characters, and other intricate features, can undergo translation with marginal reliance on human intervention. As the rules for Braille conversion vary across languages, we surmise that the ability of LLMs to recognize the language of a passage of text will be invaluable for the adaptation of multilingual documents and materials used for language instruction. Employing currently available tools, it is necessary to resort to manual pre-annotation when a document encapsulates two or more languages sharing the same script, especially when there is an expectation for a contracted Braille rendition according to the conversion guidelines of each language.

Moreover, discerning the precise pronunciation of a character is a pivotal step not only in Taiwanese Mandarin but also in Japanese Braille, Mainland Chinese Braille, and Cantonese Braille, among others. Hence, the feasibility of leveraging prompt engineering as a strategy for augmenting

Braille translation across an array of languages beckons further investigation.

VII. Conclusion

The primary objective of this research is to explore the advantages that can be derived from the utilization of prompt engineering for the automated translation of Chinese into Taiwanese Mandarin Braille.

While we observed improved character disambiguation compared to Liblouis, our approach is not suited for real-time applications as long as API calls have to be made. Additionally, the cost of API calls is another factor to consider. However, we expect that LLMs could be useful for translating longer documents for later publication by alleviating the workload on human proofreaders.

Acknowledgment

We thank Huang Wei-hao for providing us with helpful information about Taiwanese Braille. We thank Jens Watty for technical assistance with the performance evaluation and manually inspecting test results.

References

[1] C.-M. Lin and Y.-C. He, "A Study of the Resources and Services for Visually Impaired Patrons in Taiwan's Public Libraries." *Journal of Library & Information Studies*, vol. 13, no. 1, 2015.

[2] C. Egli, "Liblouis–a universal solution for Braille transcription services," in *Proceedings of Daisy 2009 Conference*, 2009.

[3] J.-X. Zhang, H.-F. Chen, B. Chen, B.-Q. Chen, J.-H. Zhong, X.-Q. Zeng *et al.*, "Design and implementation of Chinese Common Braille translation system integrating Braille word segmentation and concatenation rules," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[4] X. Wang, J. Zhong, J. Cai, H. Liu, and Y. Qian, "CBConv: service for automatic conversion of Chinese characters into braille with high accuracy," in *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, 2019, pp. 566–568.

[5] Y.-C. Chen, Y.-C. Chang, Y.-C. Chang, and Y.-R. Yeh, "g2pW: A conditional weighted softmax BERT for polyphone disambiguation in Mandarin," 2022, *arXiv:2203.10430*.

[6] S. Zhang, K. Zheng, X. Zhu, and B. Li, "A polyphone BERT for polyphone disambiguation in Mandarin Chinese," 2022, *arXiv:2207.12089*.

[7] H. Yu, W. Su, L. Liu, J. Zhang, C. Cai, and C. Xu, "Pre-training model for low-resource Chinese-Braille translation," *Displays*, p. 102506, 2023.

[8] National Taiwan Library. (2009, Oct.) 國語點字自學手冊. [Online]. Available: https://www.ntl.edu.tw/lp.asp?CtNode=759&CtUnit=256&BaseDSD=7

[9] J. Sullivan. (2023, Aug.) Early history of Braille translators and embossers. https://www.duxburysystems.com/bthist.asp.

[10] A. S. Schack and R. T. Mertz, "Braille translation system for the IBM 704," 1961. [Online]. Available: https://www.duxburysystems.org/downloads/library/history/1961_ibm.pdf

[11] D. Williams and J. Bowden, "Better Braille translation for everyone: Importance of maintaining Liblouis," 2020. [Online]. Available: https://iceb.org/2020_paper_Williams&Bowden.docx

[12] C. Egli, B. Frees, L. de Ruijter, and A. Yakuboy. (2002) Liblouis: Open-source Braille translator and back-translator. [Online]. Available: https://github.com/liblouis/liblouis

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[14] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength natural language processing in Python," 2020.

[15] "Moedict." [Online]. Available: https://www.moedict.tw/

[16] H. Chase. (2022, Oct.) LangChain. [Online]. Available: https://github.com/hwchase17/langchain

[17] C.-R. Huang and S.-K. Hsieh, "Infrastructure for cross-lingual knowledge representation-towards multilingualism in linguistic studies," *Taiwan NSC-granted Research Project (NSC 96-2411-H-003-061-MY3)*, 2010.