Transliteration of Vietnamese National Scripts into Sino-Nom scripts Using T5 Language Model

Duong Thanh Trieu[1] ✉, Dinh Si Dien[2] ✉, Tran Cong Lam Anh[1], Nguyen Ngoc Duy Tan[1],
Ngo Quang Minh[1], Nguyen Hong Buu Long[1,2] ✉

[1] Faculty of Information Technology, University of Science, Ho Chi Minh City, Viet Nam ✉ `dttrieu22@apcs.fitus.edu.vn`, `tclanh22@apcs.fitus.edu.vn`, `nndtan22@apcs.fitus.edu.vn`, `nqminh22@apcs.fitus.edu.vn`,
[2] Computational Linguistics Center, University of Science, Ho Chi Minh City, Viet Nam ✉ `dinhsidien2008@gmail.com`, ✉ `nhblong@fit.hcmus.edu.vn`

Abstract. Transliterating Vietnamese National script (chữ Quốc Ngữ) (phonetic) into Sino-Nom script (chữ Hán Nôm) (logographic) is challenging: a single syllable may correspond to many characters and disambiguation becomes harder when contextual cues are sparse or absent . Existing SMT-based tools often choose frequent but incorrect characters and perform poorly with punctuation or irregular formatting. We propose a T5-based system with two fine-tuned branches: a Vietnamese-Hán model trained on 7 million aligned classical Chinese–Vietnamese lines and a Vietnamese-Nôm model trained on 27 thousand Hán–Nôm pairs. A decision tree classifier routes inputs to the appropriate branch, while sliding-window decoding improves long-sequence handling. Post-processing with OpenCC normalizes variant forms. On 5,003 mixed Hán Nôm sequences, our system achieves BLEU 69.73 and CER 0.16, outperforming both the baseline T5 (38.74, 0.38) and the CLC tool (38.83, 0.38). These results demonstrate substantial gains in transliteration accuracy and robustness for Quốc Ngữ to Hán Nôm conversion.

Keywords: Chữ Hán Nôm (Sino-Nom script) · Chữ Quốc Ngữ (Vietnamese National Script) · Hanzi · Literature · Transliteration · Encoder-Decoder model

1 Introduction

Chữ Nôm (Nom script), an ancient Vietnamese writing system developed from chữ Hán (Hanzi), was used from the 10th to the 20th century to represent both Sino-Vietnamese borrowings and native Vietnamese words[3]. Together, chữ

[3] Nôm Foundation: What is Nôm? , last accessed 2025-02-23

Hán and chữ Nôm formed the chữ Hán Nôm (Sino-Nom script) system, which underpinned a rich literary tradition for centuries. Since its abolition, the ability to read these historical texts has almost disappeared, and learning the script requires extensive knowledge of Hanzi. Effective transliteration tools are essential for researchers and preservationists of Sino-Nom documents and provide valuable assistance for those wishing to learn this intricate writing system.

Transliterating chữ Quốc Ngữ (Vietnamese National Script) —a phonetic script—into chữ Hán Nôm —a primarily logographic script—is challenging because a single Quốc Ngữ syllable often corresponds to multiple Hán or Nôm characters with distinct meanings. Approximately 60-70% of Vietnamese vocabulary, especially in formal contexts, derives from Chinese origins[4]. For illustration, in Mandarin, 400 syllables (1,200 with tones)[2] map to over 100,000 characters[4], averaging 80 per syllable; similarly, in Vietnamese, a syllable like "ma" may map to characters like 魔("demon") or 媽("mother"). This one-to-many mapping creates severe ambiguity, particularly when context is scarce, as in short or isolated sequences.

Transformer-based encoder–decoder architectures[11] are a natural fit for this task. Originally developed for machine translation, they model long-range dependencies and can incorporate contextual cues beyond local n-gram patterns. Moreover, pre-trained Transformer models allow transfer learning in low-resource scenarios. Among these, T5 (Text-to-Text Transfer Transformer) treats every NLP task as a text-to-text problem, offering flexibility in adapting to transliteration while benefiting from large-scale pretraining[9].

In this work, we construct large-scale, high-quality parallel Quốc Ngữ and Hán Nôm corpora and develop a dual-branch T5-based system. A sequence classification module selects the appropriate branch, sliding-window decoding mitigates long-sequence errors, and post-processing with OpenCC[5] normalizes variant forms. We experiment with multiple configurations, including with/without sliding-window decoding and different fine-tuning strategies, to identify the best-performing setup. Our final system achieves BLEU 69.73 and CER 0.16 on mixed Hán–Nôm sequences, showing significant improvements over CLC tool[6] and T5 baselines and demonstrating the effectiveness of our approach for complicated, context-sensitive transliteration.

2 Related Work

Transliteration between Vietnamese National script and Sino-Nom script is a specialized area that has not received significant attention in research, primarily due to the lack of qualified datasets and the requisite prior knowledge of Hanzi.

[4] Chinese Character Variants Dictionary., last accessed 2025-02-19

[5] Conversion between Traditional and Simplified Chinese., last accessed: 2024-11-16

[6] Khoa Công nghệ Thông tin & Trung tâm Ngôn ngữ học Tính toán, Trường Đại học Khoa học Tự nhiên - ĐHQG - HCM: Kim Hán Nôm., last accesed 2024-11-16

Since the 1990s, Ngô Thanh Nhàn and Nguyễn Quang Hồng have advocated for the digitization of chữ Nôm[7], leading to the inclusion of most common Nom characters in the Unicode system.

Previous work includes the development of the Nôm Converter[8], a toolkit based on Statistical Machine Translation (SMT) [5] using the Moses framework and trained on 3,234 manually transliterated lines from 22 texts. While the converter performs well in transliterating from chữ Nôm to chữ Quốc Ngữ, it reveals weaknesses in the reverse direction. For instance, the term "**nhân**" appears in different contexts, such as "nguyên **nhân**" (原因: the reason, cause), "công **nhân**" (工人: worker), and "**nhân** từ" (仁慈: mercy), but the converter inaccurately represents all instances as the same character: 原人, 公人, and 人徐, respectively.

Another study using same approach, *"Transliterating Nôm Scripts into Vietnamese National Script using Statistical Machine Translation"*[3] improves upon previous work by using 38,897 lines from single-character dictionaries, 6,205 lines from compound dictionaries, and 7,920 sentences from parallel texts in literature, history, and religion, totaling around 370,000 monolingual corpora. However, these datasets are not perfectly aligned; up to 20% of sentence pairs have mismatched meanings or lengths. Moreover, the use of single-character dictionaries is inappropriate as they often lack sufficient context for accurate transliteration.

3 Dataset

This section details the datasets used in our research and the procedures undertaken for their processing.

3.1 Data Collection

This study used a large-scale dataset of Chinese poetry[9], comprising 853,385 poems from 29,377 different poets, ranging from the Qin Dynasty to the modern era.

Classical Vietnamese literature, dictionaries, and other corpora, sources are collected with permission from the Computational Linguistics Center (CLC) at the University of Science, Vietnam National University, Ho Chi Minh City.

3.2 Data Preprocessing

While the Chinese poetry dataset was extensive, it still has some several limitations that required preprocessing before use:

[7] Chữ nôm : Văn hoá cổ truyền và thời đại thông tin., last accessed 2025-02-22
[8] ChuNom.org: Nôm Converter., last accessed 2025-02-22
[9] Jamie Wang: Chinese poem dataset., last accessed 2024-11-16

1. **Character Conversion:** The dataset was available only in Simplified Chinese, which necessitated conversion into Traditional Chinese for compatibility with historical Vietnamese transliteration. We used OpenCC[10] for this task and further standardized variant characters based on the *First Batch of Standardized Variant Characters* (第一批异体字整理表), published by the Chinese Ministry of Culture and the Chinese Character Reform Committee[11].
2. **Segmentation:** Poems were originally stored as single-line entries without segmentation. We reformatted the dataset by splitting each poem into multiple lines at commas, periods, and removed all non-Hanzi characters.
3. **Vietnamese Transliteration:** Since the dataset lacked bilingual alignment, we generate transliterations from Sino-Nom to Vietnamese using Kim Hán Nôm tool from CLC[12].
4. **Data Cleaning & Filtering:** To ensure quality, we applied two constraints:
   - Lines in the Chinese and Vietnamese columns must contain the same number of characters.
   - Word alignment accuracy was verified using reference dictionaries.

3.3 Final Dataset

After preprocessing and validation, we created a high-quality parallel dataset consisting of over 7 million pairs of sentences from about 800,000 Chinese poems, along with more than 27,000 pairs of poems, prose, and compound dictionaries from other Sino-Nom datasets.

Table 1. Summary of Chinese Poetry dataset

Era (Chinese)	**Era (English)**	**Average length**	**Number of Sentences**
宋	Song	5.9	2,248,914
明	Ming	6.0	2,081,477
近現代-當代	Modern & Contemporary	5.8	543,545
清	Qing	5.7	445,121
唐	Tang	5.6	413,367
元	Yuan	5.8	345,471
南北朝	Southern-Northern	4.9	46,872
魏	WeiJin	4.7	37,963
金	Jin	5.9	22,030
隋	Sui	4.7	11,168
漢	Han	5.5	6,409
遼	Liao	5.7	78
秦	Qin	4.9	11
混合代	Blended Eras	5.6	817,774
Total			**7,020,190**

[10] Conversion between Traditional and Simplified Chinese., last accessed: 2024-11-16

[11] 第一批异体字整理表., last accessed 2024-11-16

[12] Khoa Công nghệ Thông tin & Trung tâm Ngôn ngữ học Tính toán, Trường Đại học Khoa học Tự nhiên - ĐHQG - HCM: Kim Hán Nôm., last accessed 2024-11-16

Based on our experiments, a test set of 5,000 entries provides a reliable performance assessment. However, evaluating multiple models with large datasets is resource-intensive. To balance accuracy and efficiency, we limited the test set to at most 1% of each sentence type, yielding over 13,000 entries.

Table 2. Summary of Sino-Nom dataset

Name (Vietnamese)	**Name (English)**	**Average length**	**Number of Sentences**
Truyện Kiều	The Tale of Kieu	7.0	3,276
Tuyển tập Hồ Xuân Hương	Ho Xuan Huong Collection	6.8	75
Chinh phụ ngâm khúc	Lament of the Soldier's Wife	7.0	35
Truyện Lục Vân Tiên	Luc Van Tien's book	7.0	92
Đại Việt Sử Ký Toàn Thư	Complete Annals of Đại Việt	16.6	15,000
Từ điển từ ghép	Compound dict	2.1	6,417
Kho ngữ liệu khác	Other corpus	6.4	2,917
Total			**27,812**

4 Metrics

Performance was evaluated using six complementary metrics that together capture character-level accuracy (handle the case of Variant Forms or Simplified Chinese instead of Traditional Chinese) and length preservation:

- **METEOR** — balances precision and recall while accounting for morphological variants through stemming and synonym matching[1].
- **BLEU** — measures n-gram precision up to 4-grams, reflecting how closely model outputs align with reference transliterations[6].
- **chrF**— computes a character n-gram F-score, making it robust to the rich morphology of Vietnamese and Chinese scripts[7].
- **Character Error Rate (CER)** — a length-normalised edit distance that offers a direct estimate of character-level transcription errors[10].
- **Minimum Edit Distance (MED)** — counts the minimal substitutions, insertions, and deletions required to transform the model output into the reference sequence[12].
- **Matched-Length Ratio** — reports the proportion of outputs whose length exactly matches the reference.

5 Model

This section describes the architecture, training strategy, and auxiliary components of our transliteration system.

5.1 Baseline model

We start from the lightweight Vietnamese T5 checkpoint released by Minh Toàn[13], whose encoder–decoder architecture comprises 6 layers, 300-dimensional

[13] t5-translate-vietnamese-nom., last accessed 2024-10-01

hidden states, 2,048-dimensional feed-forward layers, 8 attention heads, and a 30,100-token vocabulary. Pre-training on a diverse mix of modern Vietnamese prose, classical literature, and parallel Vietnamese National script - Sino-Nom script excerpts provides broad lexical coverage and reasonable character-level priors.

To gain a deeper understanding of the baseline model's performance, we evaluated it on three test sets: one in Traditional Chinese (Hanzi only), the other in Sino-Nom scipt and History of Greater Vietnam (written in Traditional Chinese). It achieved a BLEU of 46.75 (CER 0.31) on Sino-Nom, a weaker 34.97 (CER 0.43) on pure Hanzi, and a surprisingly high 94.26 (CER 0.03) on the History of Greater Vietnam dataset, which suggests overfitting.

These results highlight both the strengths and limitations of the baseline, motivating our fine-tuning strategy to improve transliteration from Quốc Ngữ to Sino-Nôm.

5.2 Multi-stage Fine-tuning

The Chinese dataset follows classical Chinese grammar and poetic conventions, while the Sino-Nom dataset consists of texts that interweave native Vietnamese words and Sino-Vietnamese borrowings, primarily written using Vietnamese grammar. However, there is a substantial imbalance in the data: the Traditional Chinese corpus contains over 7 million sentences from classical texts, whereas the Sino-Nom dataset contains fewer than 28,000 sentences.

From this point onward, the term "chữ Hán" refers to texts written in Traditional Chinese, while "chữ Nôm" refers to texts that include at least one Nôm character. To handle the distinct characteristics of each script and dataset, two branches pre-trained T5 models were fine-tuned independently: **Vietnamese-Hán Model Stage** and **Vietnamese-Nôm Model Stage**.

Fine-tuning was conducted with the following hyperparameters: a learning rate of 5×10^{-6}, batch size of 8, 3 epochs, 500 warmup steps, weight decay of 0.01, and evaluation every 1,000 steps. The model with the lowest evaluation loss was selected. These settings were used for both stages to balance dataset scale with computational efficiency, improving both transliteration accuracy and generalization.

Vietnamese-Hán model Stage: In this stage, we constructed the Hán model by fine-tuning the T5 pretrained model on over 7 million sentence pairs sourced from Chinese poetry and historical texts.

The substantial dataset contributed to its impressive performance, achieving a BLEU score of 77.75 and a Character Error Rate (CER) of 0.12 on the Hán test set, markedly outperforming T5 models and CLC model (Table 3).

Vietnamese-Nôm model Stage: In this stage, we constructed the Nôm model by fine-tuning the T5 pretrained model on approximately 27,000 sentence pairs derived from Vietnamese literature and dictionaries.

While it showed an improvement over the baseline, achieving a BLEU score of 48.93 and a CER of 0.28 on the Nôm test set, its performance gain was not as significant compared to the Hán model (Table 4). Despite this disparity, we

Table 3. Test result on Hán test set

Model	Lengths	ChrF	METEOR	BLEU	CER	M.E.D
T5 model	832/955	23.68	0.50	34.97	0.43	4.69
CLC model	870/955	23.27	0.49	34.72	0.44	4.73
Vietnamese-Hán model	**954/955**	**68.62**	**0.86**	**77.75**	**0.12**	**1.31**

Table 4. Test result on Nôm test set

Model	Lengths	ChrF	METEOR	BLEU	CER	M.E.D
T5 model	841/955	37.89	0.65	46.75	0.31	3.72
CLC model	865/955	37.89	0.64	46.90	0.32	3.74
Vietnamese-Nôm model	**868/955**	**38.81**	**0.68**	**48.93**	**0.28**	**3.41**

will incorporate the Nôm model into our final architecture instead of relying on the T5 model.

5.3 Sequence Classification

Classifying input sequences as Hán or Nôm is a necessary step in our transliteration pipeline, determining whether the Hán or Nôm model processes the input. A simple heuristic, classifying sequences containing some Vietnamese words as Nôm, is unreliable. Experimental results reveal that the Hán model retains partial knowledge of Nôm characters despite being trained on a dataset composed entirely of Traditional Chinese characters. In contrast, the Nôm model struggles with texts containing a high proportion of Hán content.

To address this, we employ a Decision Tree classifier[8] trained on diverse Hán and Nôm sequences to identify linguistic patterns for accurate classification. The classifier uses three features:

- *han*: The count of Hán words whose Vietnamese phonetic forms origined from Chinese and are rarely used in modern Vietnamese contexts (e.g., thoan, hoắc, ngoạ, ...). These are identified by subtracting the han_viet_dict from the han_dict.
- *han_ viet*: The count of Sino-Vietnamese loanwords whose phonetic forms are commonly used in Vietnamese (e.g., cộng, hoà, chính, trị, quốc, gia, ...). These words appear in han_viet_dict.
- *thuan_ viet*: The count of native Vietnamese words that are absent from both the han_dict and han_viet_dict. (e.g., tèo, tớ, tôi, vua,...)

The han_dict (2,294 words) is the set of Vietnamese phonetic sounds of Chinese character, while han_viet_dict (1,873 words) is the intersection of han_dict and a Vietnamese dictionary[14]. For instance, "**nam quốc sơn hà**

[14] Vietnamese Dictionary., last accessed: 2025-02-17

nam đế cư” yields [0,7,0] (all han_viet), and “**thoản đỉnh vô nhân mi lộc ngoạ**” yields [2,5,0] (“**thoản**” and “**ngoạ**” as han, other as han_viet).

Although not entirely straightforward, a word’s phonetic form can correspond to both Sino-Vietnamese and native Vietnamese meanings. For example, “**ngã**” may represent 我 (“I”) or the native Vietnamese meaning “to fall”. Context is key for disambiguation: in “Tôi **ngã** xuống đất” (I fell to the ground), the presence of native Vietnamese words suggests use of the Nôm model. In contrast, “**Ngã** hẫn ái nhĩ” (I deeply love you) lacking native elements, fits the Hán model. Other ambiguous cases include *đồng, yêu, thương, la, lu,* and so on.

To manage varying sequence lengths, we normalize features to proportions between 0 and 1: $[x, y, z] \rightarrow \left[\frac{x}{x+y+z}, \frac{y}{x+y+z}, \frac{z}{x+y+z}\right]$, preventing bias from longer sequences.

The Decision Tree is trained on sequences labeled Hán (0) or Nôm (1), using 5,000 Hán and 7,735 Nôm sequences, with a test set of 2,547 instances (1,000 Hán, 1,547 Nôm). It achieves high accuracy on this internal test set, 96% accuracy on 2,547 sequences. For Hán sequences, it reached 0.92 precision, 0.99 recall, and 0.95 F1; for Nôm sequences, it achieved 1.00 precision, 0.94 recall, and 0.97 F1. Furthermore, independent evaluation on pure Nôm and Hán test sets (each comprising 955 sequences) confirmed the classifier’s robust generalizability, maintaining an accuracy of 0.85 (85%) across both sets.

5.4 Addressing Long Sequence Transliteration in the Hán Model

The fine-tuned Hán model significantly improves transliteration performance but struggles with long sequences (more than 15 characters), producing inconsistent outputs like mismatched lengths, repetitive words, or irrelevant content (as show in Table 5).

Table 5. Example of Source, Target, and Hán Model Outputs

Source	**Target**	**Hán Model**
vương đô uý xuất lai kiến liễu cán nhân khán liễu lệnh chỉ tuỳ tức thượng mã lai đáo cửu đại vương phủ tiền hạ liễu mã nhập cung lai kiến liễu đoan vương	王都尉出來見了幹人看了令旨隨即上馬來到九大王府前下了馬入宮來見了端王	王都論論論論

This issue, absent in the baseline model. One possible reason for this discrepancy may lie in our fine-tuning strategy. Most of our fine-tuning data consists of sequences that are around 6 or 7 characters in length, whereas the input sequences in this case are significantly longer, at least 15 or 16 characters.

To address this, we propose a consecutive transliteration approach, splitting long sequences into smaller windows for individual transliteration, then concatenating the results. For example, the sequence “nam quốc sơn hà nam đế cư” is divided into “nam quốc”, “sơn hà”, “nam đế”, and “cư”. This preserves contextual integrity compared to traditional word-splitting methods, which may disrupt compound word like "tiên sinh" (先生: Mister, Sir,...) into "tiên" and "sinh", each of which, when standing alone, may carry multiple different meanings.

To determine the optimal window length, we tested window lengths from 1 to 15 on a 4,000-sequence test set (Table 6). These tests revealed that short windows (1–3 characters) lacked sufficient context, while excessively long windows (those larger than 12 characters) reduced performance. Window lengths of 7, 9, and 11 were found to balance context and accuracy. A window length of 7 was selected as it delivered strong performance on this main 4,000-sequence test set, achieving 3,915/4,000 matched lengths, a ChrF of 53.59, and a BLEU score of 61.21.

Table 6. Splitting approach test result on various Window lengths

Split Type	Length	ChrF	METEOR	BLEU	CER	M.E.D
No Split	1,683/4,000	30.55	0.63	43.37	0.63	20.92
Split 1	3,033/4,000	41.66	0.73	47.56	0.26	10.51
Split 2	3,938/4,000	48.70	0.78	55.66	0.21	8.49
Split 3	3,933/4,000	50.37	0.79	57.53	0.20	8.03
Split 4	3,945/4,000	51.42	0.80	58.59	0.20	7.85
Split 5	3,942/4,000	51.56	0.80	58.69	0.20	7.80
Split 6	3,908/4,000	53.25	0.81	60.41	0.19	7.47
Split 7	3,915/4,000	53.59	0.81	61.21	0.18	7.43
Split 8	3,830/4,000	53.40	0.81	60.68	0.19	7.48
Split 9	3,771/4,000	53.92	0.81	61.21	0.18	7.36
Split 10	3,724/4,000	53.67	0.81	60.76	0.19	7.44
Split 11	3,632/4,000	54.23	0.81	61.40	0.19	7.35
Split 12	3,387/4,000	53.37	0.81	60.46	0.19	7.52
Split 13	3,099/4,000	52.33	0.80	59.43	0.20	7.73
Split 14	2,828/4,000	52.09	0.80	59.27	0.20	7.81
Split 15	2,515/4,000	51.13	0.79	58.34	0.21	8.07

This setting was further confirmed on a separate, independent 1,000-sequence test set. On the second test set, the window length of 7 yielded 977/1,000 matched lengths, a ChrF of 52.30, and a BLEU score of 59.50. This consistent performance across both test sets demonstrates that this approach successfully enhances the Hán model's performance on long sequences.

5.5 Converter

In rare cases, the model may generate variant forms of Hanzi. To address these cases, our **Converter** employs the method previously described in the data preprocessing section. We utilize OpenCC[15] and the First Batch of Standardized Variant Characters[16] to ensure that the output is formatted correctly.

5.6 Complete model architecture

The Splitting approach was not applied to the Nôm model for two main reasons. First, the performance gain on the Nôm task was not significantly higher than that achieved by the Baseline and CLC models. Second, experimental results suggest that the Nôm model does not exhibit the same overfitting behavior observed in the Hán model. We hypothesize that this is due to the Nôm model being fine-tuned less extensively than the Hán model, thereby retaining its ability to handle long sequences without becoming overly stochastic or unstable.

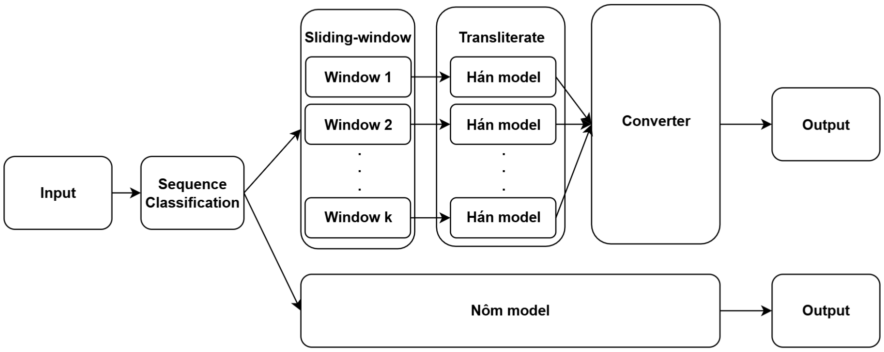

Fig. 1. Complete model architecture

6 Result

This section reports the performance of our Hán model, Nôm model, and complete dual-branch system compared with CLC model and baseline T5 models

6.1 Hán model

We test the Hán model, comparing it with the CLC and Baseline model on the test set of 12,590 Hán sequences from various periods and varieties of length from 1 to 90 that have not been use in the fine tune strategy. Table 7 shows the score of three models which show our Hán model has out-performed the current model significantly in all metrics, especially the hard case mentioned below, where it correctly transliteration all the words.

[15] Conversion between Traditional and Simplified Chinese., last accessed: 2024-11-16
[16] 第一批异体字整理表., last accessed 2024-11-16

Table 7. Test result on about 12,590 Hán poems/sequences

Model	Length	ChrF	METEOR	BLEU	CER	M.E.D
T5 model	11,056/12,590	36.45	0.55	36.64	0.40	4.49
CLC model	11,532/1,2590	36.35	0.54	36.59	0.40	4.5
Hán model with Convert and Split	**12,543/12,590**	**65.94**	**0.86**	**75.81**	**0.12**	**1.51**

A Hán test case, **"lục lục thiểu trù thất"** (碌碌少儔匹), translated as **"The mediocre are unmatched"**, from the poem "Ái Quất" by Nguyễn Bỉnh Khiêm, was selected because it poses a significant challenge: nearly all tested models failed to transliterate it correctly.

In this example, the phrase **"lục lục"** (碌碌) refers to **"the contemptible"** or **"the mediocre"**, rather than the more commonly encountered numeral **"six"** (六)[17]. Similarly, **"thất"** (匹) denotes **"to be comparable"** rather than the numeral **"seven"** (七). This illustrates a key issue in transliteration: although the model may correctly predict the phonetic sound, it may choose characters with more frequent but contextually inappropriate meanings. The phonetic forms "**lục**" and "**thất**" commonly co-occur in numerical or temporal contexts (e.g., counting, dates, or events), which likely biases the model toward interpreting them as numbers. This semantic ambiguity, compounded by contextual complexity, often leads to incorrect character selection.

6.2 Nôm model

We also evaluated the Nôm model on 955 sentences. As shown in Table 4, it offered only modest gains over the baseline and CLC tool. The main reason may because data scarcity: the Nôm corpus contains just about 28,000 pairs compared to over 7 million for Hán, limiting the model's ability to capture the diverse and ambiguous patterns of Nôm characters despite careful reprocessing.

6.3 Complete model

Finally, we evaluated the complete model on 5,000 mixed Hán and Nôm sequences of varying lengths. The results in Table 8 demonstrate a significant improvement over both the Baseline model and the CLC transliteration tool.

7 Conclusion

We presented a T5-based transliteration system for converting Vietnamese National Script (chữ Quốc Ngữ) into Sino–Nom Script (chữ Hán Nôm). To address the data imbalance between Hán and Nôm resources, we fine-tuned separate models for each script. A decision tree classifier selects the appropriate branch, while a sliding-window decoding strategy alleviates long-sequence issues in the Hán

[17] Thi Viện: Từ Điển Hán Nôm., last accessed 2025-02-20

Table 8. Test result on about 5,000 sentences mixed of Hán and Nôm

Model	Length	ChrF	METEOR	BLEU	CER	M.E.D
T5 model	4,391/5,003	33.58	0.57	38.74	0.38	4.27
CLC model	4,562/5,003	33.36	0.57	38.83	0.38	4.25
Complete model	**4,870/5,003**	**59.55**	**0.82**	**69.73**	**0.16**	**1.94**

model by reducing repetition and truncation. Combined with variant normalization, these components deliver substantial improvements over SMT-based and baseline T5 systems. Beyond its technical contributions, our approach supports the preservation, study, and teaching of Vietnamese literary heritage by enabling more accurate and robust transliteration into Sino-Nom.

Acknowledgments. The research is financially supported by Ho Chi Minh City Department of Science and Technology.

References

1. Banerjee, S., Lavie, A.: Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. pp. 65–72 (2005)
2. Chen, T.M., Dell, G.S., Chen, J.Y.: A cross-linguistic study of phonological units: Syllables emerge from the statistics of mandarin chinese, but not from the statistics of english. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 26 (2004)
3. Dinh, D., Nguyen, P., Nguyen, L.H.: Transliterating nôm scripts into vietnamese national scripts using statistical machine translation. International Journal of Advanced Computer Science and Applications **12**(2) (2021)
4. Đình Khẩn, L.: Từ vựng gốc Hán trong tiếng Việt. Nhà xuất bản Đà Nẵng (2010)
5. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., et al.: Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Companion Volume Proceedings of the Demo and Poster Sessions. pp. 177–180. Prague, Czech Republic (June 2007). https://doi.org/10.3115/1557769.1557821
6. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
7. Popović, M.: chrf: character n-gram f-score for automatic mt evaluation. In: Proceedings of the tenth workshop on statistical machine translation. pp. 392–395 (2015)
8. Quinlan, J.R.: Induction of decision trees. Machine Learning **1**(1), 81–106 (1986). https://doi.org/10.1007/BF00116251
9. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer.

Journal of Machine Learning Research **21**(140), 1–67 (2020). https://doi.org/10.48550/arXiv.1910.10683

10. Sawata, R., Kashiwagi, Y., Takahashi, S.: Improving character error rate is not equal to having clean speech: Speech enhancement for asr systems with black-box acoustic models. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 991–995. IEEE (2022)
11. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., et al.: Attention is all you need. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS 2017). pp. 5998–6008. Long Beach, CA, USA (4–9 December 2017). https://doi.org/10.48550/arXiv.1706.03762
12. Zhao, Y., Jiang, H., Wang, X.: Minimum edit distance-based text matching algorithm. In: Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering (NLPKE-2010). pp. 1–4. IEEE (2010)