IMPROVING TRANSLITERATION SYSTEM FROM NÔM SCRIPTS INTO VIETNAMESE NATIONAL SCRIPTS USING LANGUAGE MODEL

Nguyen Thi Kim Phuong¹, Nguyen Hong Buu Long¹, Dinh Dien¹, Luong An Vinh²

¹Faculty of Information Technology, University of Science, VNU-HCM
²Faculty of Information Technology, Saigon Technology University
ntkphuong@apcs.fitus.edu.vn, {nhblong, ddien}@fit.hcmus.edu.vn, vinh.luongan@stu.edu.vn

ABSTRACT: Nôm scripts were a means that our Vietnamese forefathers used to preserve our cultural identity against assimilation under periods of Chinese rule (Bắc thuộc). During roughly 1000 years from the tenth century to the early twentieth century, knowledge and experience were passed down from one generation to the next thanks to Nôm scripts as a writing system. Since the nineteenth century, Nôm scripts were gradually replaced by National scripts, which resulted in fewer and fewer Vietnamese people can read Nôm due to the difficulties in learning a different type of writing system. Nowadays, modern Vietnamese generations have lost access to 1000 years of written knowledge hidden in the ancient scripts. Fortunately, advances in computer science, namely machine translation, can help us transliterate Nôm scripts into National scripts. Therefore, we can regain access to our cultural heritage. Two major models influencing a transliteration system are the translation model and the language model. This paper proposes a method to improve a currently-in-use transliteration system called Nôm Converter. By building language models for specific literary forms and domains, together with a larger corpus, we can build a transliteration system that outperforms Nôm Converter. BLEU scores of our system are 82.80 and 89.72, while results on the same test sets of Nôm Converter are 56.84 and 50.95.

Keywords: Machine Translation, Nôm Scripts, Language Model.

I. INTRODUCTION

Transliteration is the conversion of scripts from one writing system to another. For example, we have transliteration from Korean scripts “한글” into Latin scripts “hangul”, or transliteration from Japanese scripts “せんせい” into Latin scripts “sensei”, transliteration from Vietnamese Nôm scripts “碎[]越南” into National Latin-based scripts “tôi là người Việt Nam”. In the word trans-liter-ation, the prefix trans- means “across” or “beyond”, which indicates the change, as in trans-form (to change in shape, form, or appearance), or in trans-late (to change from one language to another); liter- is a morpheme derived from “litera” which means a letter of the alphabet. So, transliteration is converting letters from one alphabet to another.

Transliteration is often conducted between different writing systems within one language as two examples of Korean-Latin scripts in the Korean language or Japanese-Latin scripts in the Japanese language. However, there are cases in which transliteration is a part of the translation process across different languages [1]. The type of transliteration considered in this study is the former one because the authors are interested in converting between scripts of only one language which is Vietnamese. Korean and Japanese transliterations are straightforward because if we have a mapping table like Table 1, we can substitute each Korean Hangul character with its corresponding Latin character(s) to get the Latin transliteration for a given Hangul script. The same goes for Japanese Hiragana scripts. The two types of transliteration in Korean and Japanese are simple because the writing systems in both cases are phonetic writing systems, and the number of mappings is reasonable for a person to learn by heart. Korean Hangul has 24 basic characters and Japanese Hiragana has 46 basic characters. On the contrary, the number of mappings from Nôm scripts to National scripts is 22,264 [2] which is too many for an average person to memorize.

Given the large number of mappings from Nôm scripts into National scripts, manual transliteration conducted by humans is costly. Therefore, we think about harnessing computing power to solve the problem. However, we analyze the dictionary [2] and find out that among 22,264 mappings, 11,610 are one-to-one mapping, and 10,654 belong to the set of one-to-many mappings. Table 2 shows some examples of two mapping-types. Nôm characters which have one mapping to National scripts are categorized as mono-transliterable. Meanwhile, Nôm characters having more than one mapping to National scripts are categorized as multi-transliterable. Multi-transliterable Nôm characters cause ambiguity in transliteration, which is the reason why computing power alone cannot solve the problem of transliterating Nôm scripts. For example, given the multi-transliterable Nôm character 体 (thây, thể) without any context, a computer cannot choose the reasonable corresponding National scripts.

Fortunately, Statistical Machine Translation (SMT) paradigm can help to solve the contextual problem thanks to the translation model and the language model. That is the reason why we apply SMT to transliterate Nôm scripts. In SMT paradigm, translation model is trained with parallel text, while language model is trained with monolingual text in National scripts. Since we can collect more monolingual text in comparison with parallel text, we choose to improve the transliteration system through language model instead of translation model.
In this paper, we present the method to improve the transliteration system from Nôm scripts into National scripts using language model. Our research steps are as the following: (1) collect and clean (i) the Nôm-National parallel corpus as the training data for the translation model and (ii) the monolingual National scripts as the training data for the language model, (2) categorize corpora based on literary forms and domains, (3) experiment, and (4) analyze the experimental results.

The remaining of the paper is organized as follows: in section II and III, we provide an overview of Nôm scripts and related work, respectively. Then, we present our proposed model in section IV and discuss the experimental results in section V. Section VI concludes the study.

### Table 1. Examples of transliteration in Korean, Japanese, and Vietnamese languages

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ㅎ</td>
<td>h</td>
<td>せ</td>
<td>se</td>
<td>碎</td>
<td>tôi</td>
</tr>
<tr>
<td>ㅏ</td>
<td>a</td>
<td>ん</td>
<td>n</td>
<td>□</td>
<td>là</td>
</tr>
<tr>
<td>ㄴ</td>
<td>n</td>
<td>せ</td>
<td>se</td>
<td>臉</td>
<td>người</td>
</tr>
<tr>
<td>ㄱ</td>
<td>g</td>
<td>い</td>
<td>i</td>
<td>越</td>
<td>Việt</td>
</tr>
<tr>
<td>ㅡ</td>
<td>u</td>
<td>南</td>
<td>Nam</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 2. Mono- versus multi-transliterable Nôm scripts

<table>
<thead>
<tr>
<th>Nôm script</th>
<th>National script</th>
<th>Quantity of National scripts</th>
<th>Nôm script category</th>
</tr>
</thead>
<tbody>
<tr>
<td>全</td>
<td>trùm</td>
<td>1</td>
<td>mono-transliteratorable</td>
</tr>
<tr>
<td>体</td>
<td>thé, thể</td>
<td>2</td>
<td>multi-transliteratorable</td>
</tr>
<tr>
<td>列</td>
<td>lệ, lôi, trói</td>
<td>3</td>
<td>multi-transliteratorable</td>
</tr>
<tr>
<td>骨</td>
<td>côt, cót, gút</td>
<td>4</td>
<td>multi-transliteratorable</td>
</tr>
<tr>
<td>點</td>
<td>châm, dém, điểm, điểm, dòm</td>
<td>5</td>
<td>multi-transliteratorable</td>
</tr>
<tr>
<td>劍</td>
<td>chém, ghem, guom, kém, kiём, sówm</td>
<td>6</td>
<td>multi-transliteratorable</td>
</tr>
</tbody>
</table>

### II. OVERVIEW OF NÔM SCRIPTS

East Asian countries were heavily influenced by Chinese culture since ancient times. Korea, Japan, and Vietnam are among those countries. All three nations first borrowed Chinese writing, then adapting and adjusting to record their own speech. In Vietnam, Chinese writing was popular in the civil service system and was used in civil service examinations during most feudal dynasties in Vietnamese history, except Ho and Tay Son dynasties. Despite its popularity, Chinese writing is not expressive enough to write all speeches in Vietnamese. As in [3], we can view each morpho-syllable as a combination of its initial, final, and tone in which initial consists of the initial consonant and the final consists of everything else, except the tone. From that perspective, there are 150 most common finals (vần) in the Vietnamese language. However, there are only 75 finals in Sino-Vietnamese system, which means Sino-Vietnamese system is not expressive enough to record all speeches in the Vietnamese language, thus there was a need to compose new scripts to address the problem and be compatible with the writing system being used at that time. Hence, Nôm scripts were created based on Chinese scripts.

Since Nôm scripts were inspired by Chinese scripts, we will review constructing methods of Chinese scripts before considering those of Nôm scripts. Radicals (部首) are basic elements that are used to construct Chinese characters, as well as to index dictionaries. Radicals are usually the semantic component of each character. For example, character 明 (bright) contains the radical 日 (sun, day, daytime). We might interpret the relationship between the radical 日 and the character 明 as “It is bright (明) during daytime (日)”. According to Han dictionary Shuowen (说文), there are six methods (六书) of forming Chinese characters, as in Table 3.

### Table 3. Six methods (六书) of forming Chinese characters

<table>
<thead>
<tr>
<th>ID</th>
<th>Chinese forming method</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pictogram (象形)</td>
<td>Chinese character derived from a picture, sometimes called hieroglyph.</td>
<td>日 (sun), 月 (moon), 山 (mountain), etc.</td>
</tr>
<tr>
<td>2</td>
<td>Ideogram (指事)</td>
<td>Chinese character indicating an idea, such as (one), 上 (above), 本</td>
<td></td>
</tr>
</tbody>
</table>
as up and down, also known as self-
explanatory character.

3 Combined ideogram (会
意) Chinese character that combines the
meanings of existing elements, also known
as joint ideogram or associate compounds. 好 (good), 明 (bright), 休
(rest), etc.

4 Ideogram plus phonetic
(形声) Phonetic compound or picto-phonetic
character, also known as phonogram. 园 (park), 拍 (to clap), 妈
(mother), etc.

5 Loan (假借) Character acquiring meaning by phonetic
association, also called phonetic loan. 沙 (sand), 背 (the back of a
body or object), etc.

6 Transfer (转注) Character with meanings influenced by
other words, sometimes called mutually
explanatory character. 老 (old), 考 (to check), etc.

According to [3], Nôm characters are created mainly by two in six methods in Table 3. Details of the two
methods and examples are presented in Table 4.

<table>
<thead>
<tr>
<th>ID</th>
<th>Nôm constructing method</th>
<th>Example</th>
</tr>
</thead>
</table>
| 1  | Ideogram plus phonetic
(形声) | 娘 (mother – /mẹ/): 女 (woman) suggests the meaning, while 美 (/měi/) provides the sound.

𠀧 (“three” – /ba/): 三 (three) suggests the meaning, while 巴 (/ba/) provides the sound. |
| 2  | Loan (假借) | 固 (/cố/): loans from Sino-Vietnamese sound of character 固 (/cố/)

主 (/chủ/): loans from Sino-Vietnamese sound of character 主 (/chủ/) |

Beside categorizing principles based on six methods of forming Chinese characters, there is another way to
categorize Nôm characters based on their structures as in Figure 1 [4].

![Diagram of Nôm categories based on structure of characters](image_url)

**III. RELATED WORK**

Through researching, we find three automatic transliteration systems that operate on different writing systems.
They are Talking Chinese to Pinyin/Zhuyin Converter, Chinese Converter, and Nôm Converter. Those systems are web
applications that allow users to transliterate Chinese or Nôm scripts into Latin-based scripts. Nôm Converter addresses
the same problem we encounter, and they also open about the toolkit used to build their system. So, we follow the same
approach as Nôm Converter [5] by applying Moses toolkit [6] to build a transliteration system. In addition, the two
Chinese Converter websites give no information on how to build the converters, so we stick with the approach
suggested by Nôm Converter. However, we only build one-direction system that transliterates Nôm scripts into
Vietnamese Nationals scripts instead of building a bi-directional system as Nôm Converter because the other direction
is not very meaningful. Moreover, by focusing on one direction, we can invest more time and resources to solve the problem of our interest.

IV. PROPOSED MODEL

In recent years, Neural Machine Translation (NMT) is preferred to Statistical Machine Translation (SMT) in order to translate high-resource language pairs. However, Nôm – National scripts is a low-resource language (script) pair, which means the amount of collected data is not sufficient for NMT. Therefore, we apply the traditional SMT approach in this study.

A. Noisy channel model

We build our transliteration system with a Software Toolkit for SMT called Moses [6]. Operations of Moses are based on the noisy-channel model in which we can combine translation model and language model to find the best transliteration output. The problem of transliterating Nôm scripts into National scripts is essentially to find the best transliteration \( q (Quốc ngữ - National script) \) for an input sentence \( n (Nôm) \)\(^1\). Consider the situation in Figure 2, suppose you have a sentence written in National scripts \( q \) and you want to send that sentence to your friend. Sentence \( q \) is passed through a noisy channel and comes out distorted. Let us call the distorted sentence \( n \). Your friend (the decoder) receives the distorted sentence \( n \) and has to reconstruct the original sentence \( q \) based on his/her knowledge about possible source sentences \( P(q) \) and the distortion \( P(n|q) \) caused by the noisy channel. Your friend can reconstruct the sentence \( p \) thanks to Bayes theorem:

\[
\hat{q} = \arg \max_q P(q)P(n|q)
\]

which is also the equation of our proposed model.

\[\text{Figure 2. Transliteration as a noisy-channel model}\]

B. N-gram language models

Language model is one of two key components in our transliteration system. Tri-gram language model is common in SMT. But based on our observation, a transliteration system with a 4-gram language model produces better results. So, we use 4-gram language model in this study. Formally, a language model is a function that takes a sentence in National scripts and returns the probability that it was produced by a Vietnamese speaker [7]. For example, a probabilistic language model \( P_LM \) should give a real sentence a higher probability than a fake one:

\[
P_{LM}(\text{tôi ăn cơm}) > P_{LM}(\text{tôi ăn sách}),
\]

or \( P_{LM}(\text{I eat rice}) > P_{LM}(\text{I eat books}) \)

This preference of the language model helps the transliteration system to choose a more natural word choice given some context (history) of the National script, that is,

\[
P_{LM}(\text{com | tôi ăn}) > P_{LM}(\text{sách | tôi ăn}),
\]

or \( P_{LM}(\text{rice | I eat}) > P_{LM}(\text{books | I eat}) \)

In the example above, "tôi ăn" ("I eat") is the history that influences the choice of the transliteration system between "com" ("rice") and "sách" ("books"). The approach to language model that take history of \( (n-1) \) previous words to predict the \( n \)-th word is the \( n \)-gram model approach. Suppose we have a full sentence \( W \) made of the sequence of words \( (w_1,w_2,...,w_n) \). A \( \text{uni-gram} \) model assigns the probability of the sentence \( W \) as the product of probabilities of each individual word in \( W \):

\[
P(W) = P(w_1)P(w_2)\ldots P(w_n)
\]

where \( P(w_i) \) is computed by the number of times of \( w_i \) seen in the training corpus over \( N \) which is the total number of words in the corpus:

\(^1\) Because both Nôm and National scripts start with letter \( n \), we use letter \( q \) to denote the output sentence in National scripts to avoid the confusion.
\[ P(w_i) = \frac{\text{frequency}(w_i)}{N} \]

For an \( n \)-gram model, we look at the previous \((n - 1)\) words to estimate the next one. For example, a 4-gram model will look at the previous three words, so the probability of the sentence \( W \) is computed as follows:

\[ P(W) = P(w_1)P(w_2|w_1)P(w_3|w_2, w_1)P(w_4|w_3, w_2, w_1) \ldots P(w_n|w_{n-1}, w_{n-2}, w_{n-3}) \]

### C. Evaluating language models: BLEU and Perplexity

In order to compare different language models to find the best one for our transliteration system, we need to measure the performance of each language model. For our transliteration task, we can use Bi-Lingual Evaluation Understudy (BLEU) to indirectly measure the contribution of a language model to the machine translation system [8]. We say “indirectly” because BLEU score measures performance of the transliteration system that consists of a language model as one of its components. That is, BLEU score does not only reflect the performance of the language model alone, but the performance of the language model along with the translation model.

Measuring BLEU score to evaluate the performance of a language model is sometimes costly because every time we want to do it, we must run the entire transliteration system end-to-end, which is inefficient in terms of time and computer resources. In such cases, we can use a metric called perplexity [9] to measure the performance of the language model independent of the transliteration system, which helps us figure out the potential performance of a language model more quickly.

BLEU score is calculated based on the transliteration output generated by our proposed model (hypothesis text in Nationals scripts) and the transliteration result that manually created by humans (reference text). More specifically, BLEU score is the geometric mean of co-occurrence of \( n \)-gram in hypothesis text and reference text. The higher the BLEU score, the better the transliteration system. Here is the popular formula of 4-gram (other \( n \)-grams can be calculated similarly):

\[
\text{BLEU} - 4 = \min \left( \frac{\text{output} - \text{length}}{\text{reference} - \text{length}} \right) \prod_{i=1}^{4} \text{Precision}_i
\]

To measure perplexity of a model, we need to have unseen text called test set that is not included in the training set or training corpus. Unlike BLEU score which is proportional to the performance of the transliteration system, perplexity inversely proportional with the performance of a language model. That is, a good language model will have low perplexity (the model is not “perplexed” or “confused”) by the test set. Let \( W \) be the test set that contains \( N \) words \((w_1,w_2,...,w_N)\), then the perplexity of a language model is calculated by the formula:

\[
PP(W) = \frac{1}{P(w_1,w_2,...,w_N)^{1/N}} = \frac{1}{\sqrt[N]{P(w_1,w_2,...,w_N)}}
\]

### V. EXPERIMENTS AND RESULTS

#### A. Data

We use a parallel corpus Nôm-National scripts to train the translation model and monolingual corpus in National scripts to train the language model. In total, the parallel corpus consists of 38,897 entries of singular dictionaries (từ điển); 6,205 entries of compound dictionary (từ điển); and 12,692 pairs of sentences from 35 literary works. For language model, monolingual data are collected from websites Gạc Sách\(^1\), Ô Cửa Sổ\(^1\), Sách Phát Giáo\(^1\), and Biên Niên Sử\(^2\). Diachronic attribute is one of the criteria we consider when collect monolingual data, beside domain and literary form. We do not collect modern data such as news, sports, entertainment, on current popular websites because Nôm scripts that need to be transliterated are written in the Middle Ages, which makes modern text not suitable to train the language model.

After gathering training corpora, we need to format them in the Moses-readable format. For parallel corpus, we first clean data by removing all punctuations and blank lines. Then we tokenize data and write tokenized text into two separate files in sentence-aligned format. That is, each Nôm sentence in the first file is corresponding to its manually transliterated National sentence in the second file. Since Nôm scripts have no distinction between lower-case and upper-case, we convert all National scripts to lower-case to reduce ambiguity. The same process is applied to the monolingual corpus, except for the sentence-alignment step.

---


In the next step, we split parallel corpus with the ratio 8:1:1 for the train:test:tune sets. The test-sets and tune-sets are further categorized into two subsets according to their literary forms and domains. The first subset contains text of verse form in Literature domain (Subset 1), the second subset contains text of prose form in Religion domain (Subset 2). Monolingual data also categorized according to literary forms and domains of Subset 1 and Subset 2. Beside Subset 1 and Subset 2, there are monolingual data of prose form in History domain (Subset 3). However, we only use Subset 3 as for comparison and not include such comparison in the Experiments section of this paper to avoid confusion. Details of monolingual data splitting are in Table 5.

<table>
<thead>
<tr>
<th>ID</th>
<th>Subset name</th>
<th>Quantity of parallel sentence pair (test, tune)</th>
<th>Quantity of monolingual sentence (train LM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Subset 1</td>
<td>1,157</td>
<td>39,675</td>
</tr>
<tr>
<td>2</td>
<td>Subset 2</td>
<td>105</td>
<td>84,381</td>
</tr>
<tr>
<td>3</td>
<td>Subset 3</td>
<td>0</td>
<td>277,784</td>
</tr>
</tbody>
</table>

B. Moses configuration file

The Moses decoder operates with setting in the Moses configuration file named moses.ini. This file is generated after we train the transliteration system with both parallel and monolingual data. However, the hyperparameters which defines the significance of the translation model and language model need to be trained with the development set, also called tuning set. The hyperparameters are different for different language (script) pairs. They are initialized randomly, so we have to fine-tune the system to attain the best results. After tuning the system, we have the optimal configuration file for operations of Moses decoder.

Both Nôm and National scripts record Vietnamese, so they have the same syntactic structures[10]. Therefore, we do not need to use the reordering model of Moses while decoding. We manually set the distortion-limit to 0 in file moses.ini to disable the reordering model, thus save time and computer resources.

C. Experiments

To evaluate the performance of our proposed model, we conduct experiments to test Subset 1 and Subset 2 against three transliteration systems:

1. Nôm Converter
2. Proposed model without additional language model (we used National scripts sentences from parallel corpus to train the language model)
3. Proposed model with additional language model

BLEU score and perplexity of those experiments are presented in Table 6. In case of Nôm Converter, we do not know if they have a language model in their transliteration system, but there is no feature on their website allows us to evaluate perplexity in case they do have a language model. Regarding BLEU score of Nôm Converter, we take Nôm scripts of Subset 1 and Subset 2 as inputs to the system. Output sentences of Nôm Converter are used to measure BLEU score of the system. Our proposed models are evaluated on both BLEU and perplexity (pp) metrics.

<table>
<thead>
<tr>
<th></th>
<th>Nôm Converter (System 1)</th>
<th>Without additional LM (System 2)</th>
<th>With additional LM (System 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subset 1</td>
<td>56.84</td>
<td>79.85 (pp: 273.8)</td>
<td>82.80 (pp: 226.3)</td>
</tr>
<tr>
<td>Subset 2</td>
<td>50.95</td>
<td>87.11 (pp: 302.8)</td>
<td>89.72 (pp: 341.0)</td>
</tr>
</tbody>
</table>

D. Analysis

Experimental results in Table 6 shows that the best transliteration system is System 3 which is the one with a language model (LM) trained on additional monolingual data, aside from the monolingual data extracted from parallel corpus. In terms of BLEU score, System 3 has the best performance on both Subset 1 and Subset 2. In terms of perplexity, LM of System 3 (LM 3) performs better than its counterpart in System 2 (LM 2) on Subset 1, but worse on Subset 2. So, a bigger LM with more training data is not always the efficient LM. Our hypothesis on the

To avoid the potential confusion, please note that LM 3 is not the same as Systerm 3, LM 3 is a component in System 3. Similarly, LM 2 is a component of System 2.
cause of the inefficiency observed in LM 3 is that introducing more data into a LM can introduce more noise which downgrades the performance of the LM. That is just our hypothesis, we need to study more to test our such hypothesis and understand factors that influence the efficiency of a LM. All in all, we still need to put a LM into a transliteration system to evaluate its performance along with the translation model, even though such evaluation is more costly than measuring the perplexity alone.

VI. CONCLUSION

In this study, we have shown similarities and pointed out differences of transliteration within a language as the substitution of alphabet characters versus across languages as a part of the translation process. We apply Statistical Machine Translation (SMT) techniques to solve the problem of transliterating Nôm scripts into National scripts. With a larger parallel corpus with LM built upon National scripts from that corpus, our proposed model outperforms the current transliteration system named Nôm Converter. BLEU scores of our proposed model in comparison with Nôm Converter are 79.85 versus 56.84, and 87.11 versus 50.95. We further improve our proposed model by building additional LMs based on literary forms and domains of training text to attain higher BLEU scores, at 82.80 and 89.72.

To further improve our transliteration system, we will continue studying to figure out factors that influence the performance of a LM and a transliteration system as a whole. That way, we can build better LMs, better transliteration systems to help Vietnamese people regain access to 1000 years of cultural heritage.

VII. ACKNOWLEDGEMENT

We would like to thank Ho Chi Minh City Department of Science and Technology for supporting us financially to conduct this study.

REFERENCES


CAI TIEN HỆ THÔNG CHUYỂN TỰ CHỦ NÔM SANG CHỦ QUỐC NGỮ BẰNG MÔ HÌNH NGỌN NGỮ

Nguyễn Thị Kim Phượng, Nguyễn Hồng Bùi Long, Đình Diện, Lương An Vinh

TÓM TẮT: Thời kỳ Bắc thuộc, các triều đại phong kiến dùng chữ Hán làm công cụ để đồng hóa dân tộc Việt Nam. Không chịu khuyết phục, chưa có tài liệu sáng tạo ra chữ Nôm để giải quyết nội, chữ viết và văn hóa dân tộc mình. Trước khi chữ Quốc ngữ trở thành văn tự chính thức của nước ta vào thế kỷ XX, chữ Nôm là phương tiện lưu giữ di sản văn hóa của người Việt trong khoảng một ngàn năm, bài đầu từ thế kỷ thứ X. Người trí thức hay chúa đều được khai thác do chỉ còn chưa tới 100 người trên thế giới có thể đọc chữ Nôm. Văn đề nên được quan tâm quan trọng, đủ điều kiện để xây dựng phương pháp cải tiến hệ thống dịch máy (chuyển tự) từ chữ Nôm sang chữ Quốc ngữ bằng mô hình ngôn ngữ. Phương pháp để xây dựng cho kết quả tính bảngBLEU đạt 82.80 và 89.72, cao hơn hạn Nôm Converter với mức điểm tương Ứng là 56.84 và 50.95 trên hai tập dữ liệu đánh giá.