Statistical Romanization for Abugida Scripts:
Data and Experiment on Khmer and Burmese

Chenchen Ding¹ Vichet Chea² Win Pa Pa³ Masao Utiyama¹ Eiichiro Sumita¹
¹ASTREC, NICT, Japan ²NIPTICT, Cambodia ³UCSY, Myanmar
¹{chenchen.ding, mutiyama, eiichiro.sumita}@nict.go.jp
²vichet.chea@niptict.edu.kh ³winpapa@ucsy.edu.mm

1 Introduction

Linguistically, romanization is the task of transforming a non-Latin writing system into Latin script. It is a language-specific task in natural language processing (NLP). Despite standard romanization systems, conventional transcription methods are applied prevalently in many languages. The spellings are thus inconsistent and complicated in some cases. Consequently, statistical approaches are required for an efficient solution on this task.

We focus on the name romanization for languages using abugida scripts. Different from an alphabetic system, in an abugida system, consonant letters stand for a syllable with an implicit inherent vowel, and diacritics modify consonant letters to change or depress the inherent vowel. We designed annotation schemes to establish a precise, consistent, and monotonic correspondence between the two different writing systems on grapheme level, through which various machine learning approaches are facilitated.

Specifically, we collect and manually align 7,658 and 2,335 person name romanization instances for Khmer and Burmese (Myanmar), respectively. Both languages are with limited resources and NLP studies. Experimental results demonstrate that standard approaches of conditional random fields (CRF) and support vector machine (SVM) supervised by the manually annotated data achieve high precision in romanization. The annotated data have been released under a CC-BY-NC-SA license.¹

2 Related Work

In engineering practice, the romanization task is a string-to-string transformation, which can be cast as a simplified translation task working on grapheme level rather than on word (or phrase) level with no (or few) reordering operations. Hence, general SMT techniques can be facilitated once training data are prepared. The phrase-based SMT plays a role of baseline in recent workshops [1, 2], whereas neural network techniques provide further gains in performance [4, 5]. Although neural network-based, pure string-to-string approaches prove powerful on different transliteration tasks, there is still room for improvement, especially on tasks between different writing systems. For example, the Thai-to-English task, which is similar to our task, has relatively poor performance in NEWS 2015. The problem around diacritics in abugida is also stated in Kunchukuttan and Bhattacharyya [6]. General techniques may offer acceptable solutions overall, but specialized investigation and processing are required for further improvement on tasks for specific languages. In the remainder of the paper, the Khmer and Burmese data with manual annotation are described in Secs. 3 and 4, respectively. Experiments are reported in Sec. 5, and Sec. 6 contains the conclusion and future work.

3 Khmer Data

We collect and annotate 7,658 real Khmer name romanization instances. An example of a Khmer name aligned with its romanization is illustrated in Fig. 1.

A special feature of Khmer script is that there are two series of consonant letters, which have different inherent vowels. Furthermore, diacritics represent different vowels when added to corresponding consonant letters, which leads to a very complicated vowel system. Another feature of Khmer script is that the vertically stacked consonant letters are very common. One reason is the abundant consonant clusters in phonology, another reason is the etymological spelling of Sanskrit (Pali) derived words. Those stacked letters are not strictly based on the syllable structure. Additionally (and more problematically), the virama, i.e., the diacritic used to suppress the inherent vowel, is absent in Khmer script, which makes the identification of onset and coda difficult.

¹The Khmer data are available at http://niptict.edu.kh/khmer-name-romanization-with-alignment-on-grapheme-level/. The Burmese data are available at http://www.nlpresearch-ucsy.edu.mm/NLP_UCSY/name-db.html. Please contact the first author to obtain the data if the servers are unstable.
As phonemic syllables and writing units do not match well in Khmer script. Consistent alignment principles are thus difficult to establish if the syllables or writing units are taken as atoms in processing. Furthermore, there are numerous types of syllables and writing units, which are too complex for statistical model learning because of sparseness. Based on these facts, we prefer a pure character-level alignment, where standalone consonant letters, diacritics, and the invisible staking operator are separated and treated equally as grapheme on the Khmer side.

A consequent problem is the inherent vowels, once we completely ignore the syllable structure. We cannot judge whether a standalone consonant stands for only one consonant or contains a further inherent vowel, due to ambiguity in the writing system. We thus apply a scheme to insert a mark to represent the inherent vowel for all the “bare” consonant letters (i.e., consonant letters without any diacritics or stacking operators) to establish a consistent alignment. This insertion is thus decisive based on the surface spelling, and the ambiguity on the presence or absence of the inherent vowel is converted to whether the inserted mark is silent.

In the example of Fig. 1, the consonant letters are 1, 3, 4, 6, 8, and 10, where 3 and 4 are bare consonant letters,2 after which the inherent vowel is inserted (noted by a dot here and indicated by a wedge without original index). Then a character-by-character alignment can be established.3 According to our overall principles, Khmer consonant letters are aligned to Latin consonant graphemes; diacritics, including inserted inherent vowels, are aligned to Latin vowel graphemes; and any silent part is aligned to a placeholder.

4 Burmese Data

We collect and annotate 2,335 real Burmese name romanization instances, including students and faculty names in a university, public figure names and names from minority nationalities in Myanmar.

Syllable are clear and integrated units in Burmese script, which can be identified by rules, i.e., all diacritics and consonant letters with a virama must be attached to the letter they modify [3]. As an example, a Burmese syllable composed of six characters is illustrated in the beginning of the section, where the numbers indicate the order of the characters in the composition. 1 and 4 are consonant letters, and 2, 3, 5, and 6 are diacritics. Specifically, 2 and 3 form consonant clusters with 1 as a tone mark, and 6 is the virama to depress the inherent vowel of 4 to form the coda of the syllable.

In order to provide an efficient interface for the Romanization task, we step into the syllable structure and further extract sub-syllabic segments, i.e., onset, rhyme, and tone, to achieve a precise and consistent correspondence between Burmese and Latin scripts. Fig. 2 is a diagram of the sub-syllabic segmentation scheme. It is relatively intuitive to divide a Burmese syllable into two parts, onset and rhyme, despite the difficulties introduced by medial consonants. The rhyme is not further dividable except in to stripe tones annotated by explicit marks.

Specifically, four diacritics are used to represent the medial consonants, i.e., yapin, yayit, wahswe, and hahto to represent /-j-/,-j/-, /-w-/, and /-w-/, respectively. As shown in Fig. 2, yapin, yayit, and hahto are placed into the onset part, but wahswe into the rhyme part. This scheme is adopted for two reasons. (1) The phonotactical constraints are strict on yapin, yayit, and hahto, but loose on wahswe. Hahto is only used on sonorants (nasals and approximants) and yapin / yayit only on velars and bilabials.6 In contrast, wahswe can be freely combined with all consonant phonemes with the triv-

---

2 As the task is to transform Khmer script to Latin script, the graphemes are not guaranteed to be single letters on the romanization side, e.g., the 5 corresponds to Ω here.
3 Yapin originally represents /r/- while in the modern standard Burmese the phoneme /r/ has been merged into /j/.
4 Actually a voiceless sign, e.g., changing /w/- to /j/.
5 yapin can also be combined with /r/-.
Figure 2: Burmese sub-syllabic segmentation.

coda* +
+ medial
nu-

initial syllable

syllabic

while

syllable

Figure 3: Transposition before segmentation.

A syllable can be divided into a nucleus and its coda. The nucleus contains the main vowel of a syllable, while the coda consists of the consonants that follow the nucleus. The initial consonant is the consonant that appears before the nucleus.

1. The combination of /w/ may be argued in some references. The combination appears marginally in borrowing words and interjections.
2. The visarga (high) and aukmyit (creaky) are segmented in our scheme, as they are “pure” tone marks.

5. Experiment

We tested a state-of-the-art bi-directional LSTM-based RNN approach [8], [23] as well as standard CRF and SVM approaches on our data.

The experiments were cross-validated. The RNN handles the task in a sequence-to-sequence way on character-level, without using any a priori knowledge, while CRF and SVM take advantage of the designed segmentation and manual alignment. The features for CRF and SVM were tokens up to trigrams. The settings from the original paper were used for the RNN. We evaluated the experimental results by two metrics: the accuracy of target tokens and the accuracy of target strings (LAT) where BLEU [10] on Latin letters was used.

The experiments by LSTM-based RNN are conducted using an eight-fold cross-validation on our data without using manual alignment. The performance reaches .953 in terms of LAT on our Khmer

---

7e.g., /kʃ/ is actually /c/ or /kʃ/.
8e.g., changing /a/ to /a/ and changing /-as/ to /-us/.
9The visarga is usually not transcribed and aukmyit is inconsistently represented by a final ʃ in Romanization.
10Multiple medial consonants for one initial consonant is possible while yapin and yayit cannot appear simultaneously.

---

12As mentioned, glottal endings take no tones.
13However, the swapped order may introduce no problem in displaying, so both orders are used in daily typing.
14https://github.com/1maooli/Agtarbidir
17Embedding size is 500, hidden unit dimension is 500, and batch size is 4. AdaDelta is used for optimization with a decay rate ρ of 0.95 and an ϵ of 10−6.
18TK cannot be applied to the RNN approach as the alignment is not an explicit variable.
data, which is an acceptable result. While RNN approach cannot performed well on our Burmese data, where LAT is no more than .718. We find the RNN actually generates “Burmese-styled” Latin transcriptions but inaccurate. We thus consider the performance to be reasonable and attribute the causes for the result to (1) the data size, which is still insufficient to support an RNN model (less than one third of the Khmer data), and (2) the mapping between Burmese and Latin scripts is complex, where many-to-many alignment between characters is common and is difficult to model without any heuristics.

Similar to RNN experiments, CRF and SVM experiments are also cross-validated, where the eight-, four-, and two-fold results are tested for comparison. The results by CRF and SVM on Khmer and Burmese data are listed in Tables 1 and 2, respectively. The performance of the two approaches are similar: TOK is around .99 and LAT is around .98 on Khmer data; TOK is around .95 and LAT is around .91 on Burmese data. The performances outperform those of RNN on both data sets, respectively. We conclude that the romanization task does not require features in a long distance, which RNN can model well, but precise local alignment provides useful and efficient information contributing to the performance. Therefore, we consider that it is the high quality of our annotated data, rather than a sophisticated model, that contributes more to the task.

As to engineering issues in practice, it takes hours to train an RNN model on several thousand instances in eight-fold cross-validation, while to train the SVM model in KyTea only takes seconds. Therefore, we consider RNN to be a superfluous approach for the Khmer romanization task.

<table>
<thead>
<tr>
<th></th>
<th>2-fold</th>
<th>4-fold</th>
<th>8-fold</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOK</td>
<td>.987 / .988</td>
<td>.988 / .989</td>
<td>.989 / .990</td>
</tr>
<tr>
<td>LAT</td>
<td>.974 / .977</td>
<td>.976 / .978</td>
<td>.977 / .979</td>
</tr>
</tbody>
</table>

Table 1: Results on Khmer data.

<table>
<thead>
<tr>
<th></th>
<th>2-fold</th>
<th>4-fold</th>
<th>8-fold</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOK</td>
<td>.946 / .944</td>
<td>.948 / .947</td>
<td>.947 / .947</td>
</tr>
<tr>
<td>LAT</td>
<td>.912 / .907</td>
<td>.913 / .911</td>
<td>.913 / .910</td>
</tr>
</tbody>
</table>

Table 2: Results on Burmese data.

6 Conclusion and Future Work

We focused on the Romanization tasks on Khmer and Burmese, from the data preparation to experiments and discussion of statistical approaches. The data prepared in this study have been released under a CC-BY-NC-SA license to promote the research of low-resourced language processing.

Acknowledgements

We thank Dr. Lemao Liu for his help in the RNN experiments. We thank Mr. Kaing Hour for his help in checking the Khmer data.

References