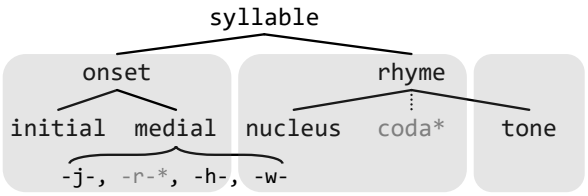

Figure 2: Burmese sub-syllabic segmentation.

ial exception of /w-/.[7] (2) *Yapin*, *yayit*, and *hahto* may change the property of the initial consonants while *wahswe* may change the property of the nucleus vowels. Besides the voiceless marker of *hahto*, *yapin / yayit* palatalize velar consonants.[8] Contrarily, *wahswe* may add a rounded feature to the following nucleus vowels.[9] The two issues affect conventional romanization spelling that multigraphs of Latin letters are used to transcribe onset clusters with *yapin*, *yayit*, and *hahto*, or *wahswe*-rhyme clusters. So, the scheme is the only feasible way to segment Burmese onset and rhyme to achieve a monotonic and one-to-one and alignment with Latin script in conventional romanization.

Burmese has two codas, nasal and glottal, and three tones, creaky, low, and high. The three tones can be combined freely for open and nasal-ended syllables, but not for glottal ended ones. The glottal ending thus may be regarded as a fourth tone in some analysis. Further, the nasal ending can be regarded as a nasalization of nucleus vowels. Hence, the coda is not a necessary component from an extreme viewpoint, which places the nasal ending into nuclei and the glottal into tones. However, in the writing system, the nasal and glottal endings are represented by consonant letters with a virama. These "coda-letters" affect nucleus vowels, so that they are not completely detachable. As a result, only two tone marks, i.e., the visarga (high) and *aukmyit* (creaky) are segmented in our scheme, as they are "pure" tone marks,[10] and any further segmentation would ruin the integrated of the rhyme.

Specific transposition is required for the sub-syllabic segmentation. An examples is illustrated in Fig. 3. For the onset-rhyme segmentation, an onset part with multiple components is coded in the order of "C + *yapin / yayit* + *wahswe* + *hahto*", where C is the initial consonant.[11] It is obvious that the segmentation cannot be conducted under the coding order once *wahswe* and *hahto* appear simultaneously. Hence, a ***wahswe-hahto* swapping** is required. For the rhyme-tone segmentation, one problem is the order of the *aukmyit* and virama in nasal-ended creaky-toned syllables.[12] As the standard order should be "*aukmyit* + virama" in coding,[13] an ***aukmyit*-virama swapping** is required.

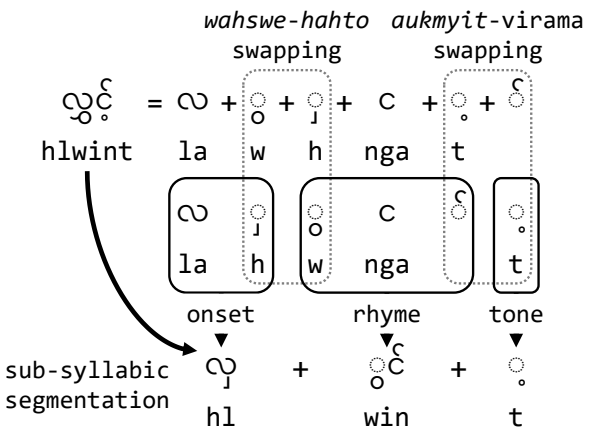

Figure 3: Transposition before segmentation.

In addition to the transposition pre-processing, a "dummy rhyme" is inserted for all the "bare onsets", identical to the insertion process applied on Khmer.

5 Experiment

We tested a state-of-the-art bi-directional LSTM-based RNN approach [8],[14] as well as standard CRF[15] and SVM[16] 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.[17] We evaluated the experimental results by two metrics: the accuracy of target token labeling (`TOK`)[18] 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

[7] /ʔw-/ may be argued in some references. The combination appears marginally in borrowing words and interjections.

[8] e.g., /kj-/ is actually /c-/ or /tɕ-/

[9] e.g., changing /-aʔ/ to /-ʊʔ/ and changing /-aɴ/ to /-ʊɴ/

[10] The visarga is usually not transcribed and *aukmyit* is inconsistently represented by a final `t` in Romanization.

[11] Multiple medial consonants for one initial consonant is possible while *yapin* and *yayit* cannot appear simultaneously.

[12] As mentioned, glottal endings take no tones.

[13] However, the swapped order may introduce no problem in displaying, so both orders are used in daily typing.

[14] `https://github.com/lemaoliu/Agtarbidir`

[15] **CRF++** [7, 11] at `http://taku910.github.io/crfpp/`

[16] **KyTea** [9] at `http://www.phontron.com/kytea/`

[17] Embedding 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}.

[18] `TOK` cannot be applied to the RNN approach as the alignment is not an explicit variable.

	2-fold	4-fold	8-fold
TOK	.987 / .988	.988 / .989	.989 / .990
LAT	.974 / .977	.976 / .978	.977 / .979

Table 1: Results on Khmer data.

	2-fold	4-fold	8-fold
TOK	.946 / .944	.948 / .947	.947 / .947
LAT	.912 / .907	.913 / .911	.913 / .910

Table 2: Results on Burmese data.

data, which is an acceptable results. 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.

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

[1] Rafael E. Banchs, Min Zhang, Xiangyu Duan, Haizhou Li, and A. Kumaran. Report of NEWS 2015 machine transliteration shared task. In *Proc. of NEWS*, pages 10–23, 2015.

[2] Marta R. Costa-jussà. Moses-based official baseline for NEWS 2016. In *Proc. of NEWS*, pages 88–90, 2016.

[3] Chenchen Ding, Ye Kyaw Thu, Masao Utiyama, and Eiichiro Sumita. Word segmentation for Burmese (Myanmar). *ACM Transactions on Asian and Low-Resource Language Information Processing*, 15(4):22, 2016.

[4] Andrew Finch, Lemao Liu, Xiaolin Wang, and Eiichiro Sumita. Neural network transduction models in transliteration generation. In *Proc. of NEWS*, pages 61–66, 2015.

[5] Andrew Finch, Lemao Liu, Xiaolin Wang, and Eiichiro Sumita. Target-bidirectional neural models for machine transliteration. In *Proc. of NEWS*, pages 78–82, 2016.

[6] Anoop Kunchukuttan and Pushpak Bhattacharyya. Data representation methods and use of mined corpora for Indian language transliteration. In *Proc. of NEWS*, pages 78–82, 2015.

[7] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*, pages 282–289, 2001.

[8] Lemao Liu, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. Agreement on target-bidirectional LSTMs for sequence-to-sequence learning. In *Proc. of AAAI*, pages 2630–2637, 2016.

[9] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proc. of ACL-HLT*, pages 529–533, 2011.

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318, 2002.

[11] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proc. of HLT-NAACT*, pages 134–141, 2003.