Burmese (Myanmar) Name Romanization: A Sub-syllabic Segmentation Scheme for Statistical Solutions

Chenchen Ding[1(✉)], Win Pa Pa[2], Masao Utiyama[1], and Eiichiro Sumita[1]

[1] Advanced Translation Technology Laboratory, ASTREC,
National Institute of Information and Communications Technology,
3-5 Hikaridai, Seikacho, Sorakugun, Kyoto 619-0289, Japan
{chenchen.ding,mutiyamam,eiichiro.sumita}@nict.go.jp
[2] Natural Language Processing Lab, University of Computer Studies,
Yangon, Myanmar
winpapa@ucsy.edu.mm

Abstract. We focus on Burmese name Romanization, a critical task in the translation of Burmese into languages using Latin script. As Burmese is under researched and not well resourced, we collected and manually annotated 2,335 Romanization instances to enable statistical approaches. The annotation includes string segmentation and alignment between Burmese and Latin scripts. Although previous studies regard syllables as unbreakable units when processing Burmese, in this study, Burmese strings are segmented into well-designed sub-syllabic units to achieve precise and consistent alignment with Latin script. The experiments show that sub-syllabic units are better units than syllables for statistical approaches in Burmese name Romanization. The annotated data and segmentation program have been released under a CC BY-NC-SA license.

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), and an important module in a statistical machine translation (SMT) system, required to handle the unknown proper nouns that occur when translating a language using non-Latin script into languages using Latin script.

The task of Burmese (Myanmar) name Romanization is investigated in this study. Burmese is an understudied language and there are not many resources available for its study, although research on Burmese begun in recent years. Current research on Burmese focuses on applying available techniques to perform basic tasks in NLP such as tokenization [5], parsing [4], and SMT [3], as well as to perform basic tasks in speech processing, such as automatic speech recognition [10] and speech synthesis [17].

This study thus proceeds from raw Romanization instance collection to manual annotation and finally empirical investigation. The annotation was conducted

K. Hasida and W. P. Pa (Eds.): PACLING 2017, CCIS 781, pp. 191–202, 2018.
https://doi.org/10.1007/978-981-10-8438-6_16

considering Burmese phonology and phonotactics, as well as the conventional Romanization spellings. Statistically based experiments were then performed on the annotated data. The contribution of this study is that well-designed sub-syllabic units rather than integrated syllables in Burmese, are regarded as segments and proved to be efficient representations through experimental results. Compared with syllables, these sub-syllabic units, although they need a specific process to extract, reduce the number of segment types, which can significantly reduce the sparseness for statistical approaches. The annotated data and a `python` implementation of the sub-syllabic segmentation scheme have been released for NLP community under a `CC BY-NC-SA` license.[1]

2 Related Work

The Romanization task is a string-to-string transformation that can be cast as a simplified translation task working on grapheme level rather than word (or phrase) level with no (or few) reordering operations. Thus, general SMT techniques can be facilitated once training data are available. Phrase-based SMT plays an important role in Romanization tasks and has been taken as a baseline system in recent workshops [1,2], whereas recent neural network techniques provide further gains in performance [6,7].

Although there are several available transcription guidelines for Burmese Romanization,[2] conventional and inconsistent spellings are more prevalent in practice use. To the best of our knowledge, there is still no systematic research on this specific topic in NLP. Related studies may include the grapheme-to-phoneme (G2P) study on Burmese [16], which is a task for speech processing rather than NLP, and thus relatively large units in utterance, i.e., syllables, were used in this research. Compared with G2P, the Romanization task further suffers from the diversity of conventional spellings. For example, a common Burmese rhyme /-iɴ˥/ may have four equal variants `-in`, `-inn`, `-yn`, and `-ynn` in practice. Non-phonological spellings are also common, such as representation of a creaky tone by an ending `t` or using English-like spellings for similar phonemes, e.g., `-ike` for /-aiʔ/, and `-oo` for /-u˥/.

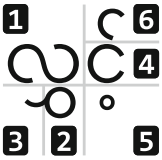
As a primary contribution of this study, we provide a solid basis for the task of Burmese name Romanization, for the first time, that includes all components from data to practical techniques. We use sub-syllabic units, specifically, onset, rhyme, and tone, and experimentally prove that they are more flexible and precise units than syllables for statistical approaches. In a wider sense, as the study is based on an insightful consideration of Burmese phonology and phonotactics, we believe the sub-syllabic units are also applicable for other related Burmese processing tasks.

[1] http://www.nlpresearch-ucsy.edu.mm/NLP_UCSY/name-db.html.

[2] Typical ones are the Myanmar Language Commission Transcription System, the Library of Congress' ALA-LC Romanization index system for Burmese (http://www.loc.gov/catdir/cpso/romanization/burmese.pdf), and the Okell's system [13].

3 Burmese Syllable

3.1 Syllable Composition



Burmese applies an abugida writing system, i.e., consonant letters stand for a syllable with an implicit inherit vowel, and diacritics modify consonant letters to change/depress the inherent vowel, to form consonant clusters, or to change tones. A Burmese syllable 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. 2 and 3 form consonant clusters with 1. 5 is a tone mark. 6 is the virama to depress the inherent vowel of 4 to form the coda of the syllable. As illustrated, Burmese syllables are usually composed of multiple characters, while syllables can be identified by rules, i.e., all diacritics and consonant letters with a virama (inherent vowel depressor) must be attached to the letter they modify [5].

3.2 Syllable-Based Processing and Limitation

Syllables are usually treated as basic atoms, i.e., unbreakable units, in related NLP studies such as Burmese word segmentation [5] or SMT [3]. In these studies, the syllable identification is the first step before any further processing. This treatment is proper regarding to the nature of those morphology or syntax related NLP tasks, as structures and fragments within syllables generally have no significant meaning attributing to sentence-level analysis.

Syllable-based processing is also a straightforward way in the Romanization task. However, the structure within syllables have a clearer and more important role in the task, because Romanization is a task more related to phonological and phonotactic features. On the other hand, the syllable turns to be a too integrated unit for detailed analysis and modeling in terms of phonology, where different parts within a syllable should be reviewed. To solve the Romanization task more efficiently, we step into the syllable structure and extract sub-syllabic segments, to achieve a precise and consistent correspondence between Burmese and Latin scripts by establishing a monotonic and one-to-one alignment.

An intuitive example on the proposed sub-syllabic segmentation is illustrated in Fig. 1, with the comparison of syllables and characters. As illustrated, syllables are integrated units but with relatively complex internal structures, i.e., syllables are too large segments in processing; character-level analysis, oppositely, turns too detailed that the correspondence between the two writing systems are difficult to sort out, i.e., characters are too tiny segments. In practice, the syllable-based processing may cause serious sparseness in model learning due to the large amount of different types, and the character-based processing introducing excessive and tricky alignment. The proposed sub-syllabic segmentation is thus at a good balance point of integrity and preciseness between the two extremes in handling the Romanization task.

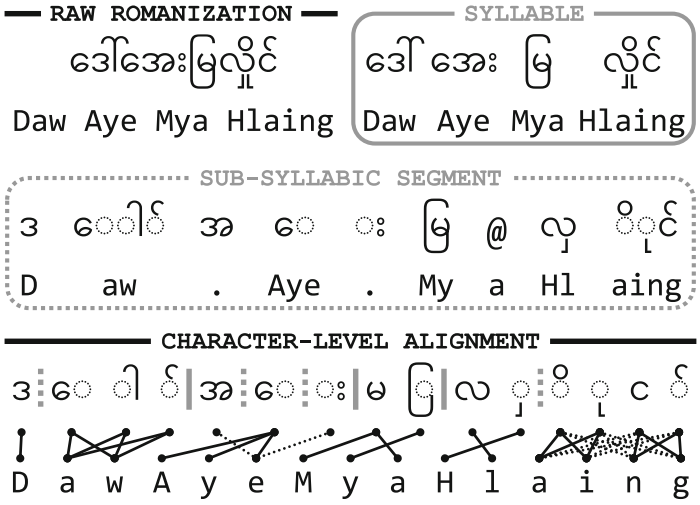


Fig. 1. Overview of sub-syllabic segmentation. Upper-left is a raw Romanization instance; upper-right shows syllables. In the sub-syllabic segmentation, ◌/. show inserted/silent segments. The character-level alignment is complex, where dash lines show the spellings affected by surrounding Burmese characters. Boundaries of syllables and sub-syllabic units are illustrated by solid and dash vertical bars, resp., in the character-based analysis.

As syllables are integrated units in the Burmese script, the segmentation scheme requires further transposition and insertion processing. The descriptions of segmentation and the transposition/insertion processing are given in the following subsections respectively.

4 Sub-syllabic Segmentation

4.1 Onset-Rhyme-Tone Segmentation

Figure 2 is a diagram of the overall sub-syllabic segmentation scheme, where the three gray blocks are the sub-syllabic units designed in this study. It is relatively intuitive to divide a Burmese syllable into two segments, onset and rhyme, despite the difficulties introduced by medial consonants, where multigraphs of Latin letters may be used for Burmese letter clusters (Fig. 3). The rhyme is not further dividable except to stripe tones annotated by explicit marks.

Specifically, four diacritics are used to represent the medial consonants, i.e., ***y****apin*, ***y****ayit*, ***w****ahswe*, and ***h****ahto* to represent /-j-/, /-j-/,[3] /-w-/, and /-^{h}-/,[4] respectively. As shown in Fig. 2, *yapin*, *yayit*, and *hahto* are placed into the onset

[3] *Yayit* 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 /n-/ to /n̥-/.

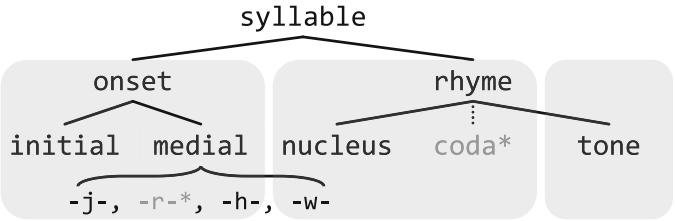


Fig. 2. Proposed sub-syllabic segmentation.

segment, but *wahswe* into the rhyme segment. 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.[5] In contrast, *wahswe* can be freely combined with all consonant phonemes with the trivial exception of /w-/.[6] (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.[7] Contrarily, *wahswe* may add a rounded feature to the following nucleus vowels.[8] These two issues affect conventional Romanization spelling. As shown in Fig. 3, multigraphs of Latin letters are used to transcribe onset clusters with *yapin*, *yayit*, and *hahto*, or *wahswe*-rhyme clusters. Consequently, our scheme is the only feasible way to segment Burmese onset and rhyme to achieve a monotonic and one-to-one alignment with Latin script in conventional Romanization.

Burmese has two codas, i.e., nasal and glottal, and three tones, i.e., 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,[9] and any further segmentation would ruin the integration of the rhyme.

[5] *Yapin* can also be combined with /l-/.
[6] /ʔw-/ may be argued in some references. The combination appears marginally in borrowing words and interjections.
[7] E.g., /kj-/ is actually /c-/ or /tɕ-/.
[8] E.g., changing /-aʔ/ to /-ʊʔ/ and changing /-aɴ/ to /-ʊɴ/.
[9] The visarga is usually not transcribed and *aukmyit* is inconsistently represented by a final `t` in Romanization.

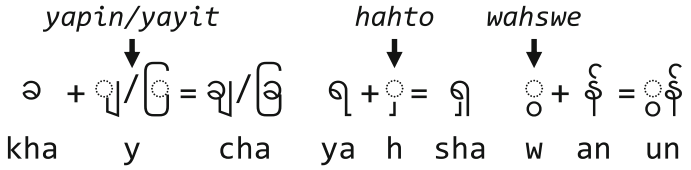


Fig. 3. Examples of common Latin multigraphs used for Burmese letter clusters.

4.2 Transposition and Insertion

Although the sub-syllable units have been figured out in the previous subsections, a raw Burmese string requires pre-processing for the segmentation, due to the coding order of Burmese script. Examples of segmentation with transposition and insertion are shown in Figs. 4 and 5, respectively.

For the onset-rhyme segmentation, an onset segment with multiple components is coded in the order of "C + *yapin/yayit* + *wahswe* + *hahto,*" where C is the initial consonant.[10] 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.[11] As the standard order should be "*aukmyit* + virama" in coding,[12] an ***aukmyit*-virama swapping** is required.

In addition to the transposition pre-processing, a "dummy rhyme" is inserted for all the "bare onsets", i.e., syllables without an explicit rhyme where the implicit inherent vowel performs the role of the nucleus. This insertion post-processing can provide a precise alignment between Burmese and Latin scripts, as the vowels are always explicit in an alphabetic system but may be implicit in an abigida system. Note that the example in Fig. 4 and the final example (`mwa`) in Fig. 5 do not require this insertion because the rhyme segments are not empty, i.e., "coda-letter" and *wahswe* can perform the role of the rhyme segment.

5 Annotated Data

Our Burmese Romanization instances were first collected from students and faculty names in a university. More instances were then collected from the Internet, including names of public figures and names from minority nationalities in Myanmar.

To process the raw instances, we first segmented Burmese and Latin strings roughly into consonant and vowel segments and used a greedy algorithm to

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

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

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

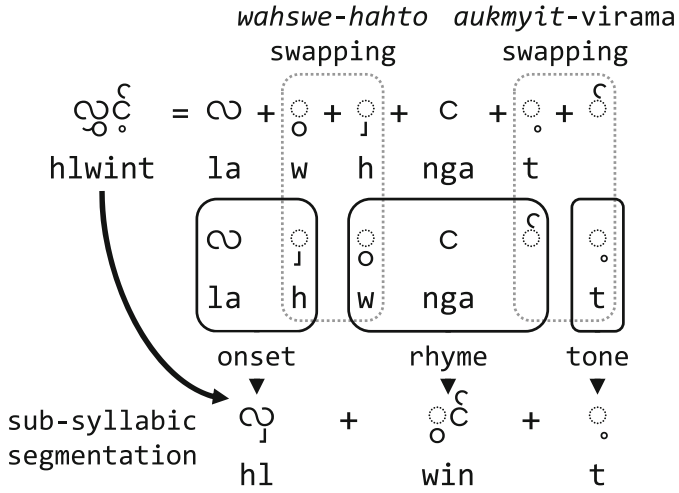


Fig. 4. Transposition before segmentation.

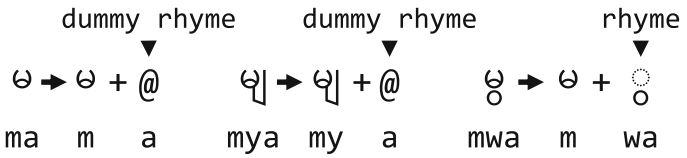


Fig. 5. Insertion for onsets without a rhyme.

generate a preliminary alignment with the help of an aligner.[13] Around 60% of the instances were aligned consistently in this stage, and we manually checked the errors to refine the overall segmentation scheme. In the second round, we filtered out questionable instances by cross-checking. Errors were found and cleaned in around 10% of the instances. In the third round, more tolerable instances, i.e., Burmese names with more than one acceptable Romanization, were added. We obtained a final total of 2,335 annotated instances.

Specific annotation samples are illustrated in the appendix attached to this paper.

6 Experiment

In recent research, direct sequence-to-sequence approaches launched by neural network techniques have been widely applied in various NLP tasks. We experiment a state-of-the-art long short-term memory (LSTM) based recurrent neural network (RNN) approach with a bidirectional search in decoding[14] [9]. The approach performs well on different transliteration tasks. We also test two standard

[13] Using **GIZA++** [12] at http://www.statmt.org/moses/giza/GIZA++.html.
[14] An open-sourced tool is available at https://github.com/lemaoliu/Agtarbidir.

machine learning approaches, conditional random fields (CRF) and support vector machine (SVM), by handling the task in a sequence labeling manner. We use the **CRF++** toolkit[15] [8,15] and the **KyTea** toolkit[16] [11] for CRF and SVM experiments, respectively.

All 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,[17] while CRF/SVM take advantage of the designed segmentation and manual alignment. The features for CRF/SVM were segments up to tri-grams. Specifically, the `-charn` and the `-charw` options are set to three for **KyTea**. The features used for **CRF++** are C_n^{n+k} $(k \in [1,2], n \in [-k,0])$ for segment sequences and C_n $(n \in [-2,2])$ for single segments. The settings from the original paper were used for the RNN: 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}. We evaluated the experimental results using two metrics: the accuracy of target segments (`SEG`),[18] and the accuracy of target strings (`STR`) where the BLEU score [14] at Latin letter level was applied.

The generalized LSTM-based RNN approach cannot handle the task well on our data. The `STR` is around .72 in 8-fold cross validation. The RNN actually generates "Burmese-styled" Latin transcriptions but inaccurate. We thus consider the performance to be reasonable and attribute the causes to (1) the data size, which is insufficient to support an RNN model, and (2) the intrinsically complex mapping between the two writing systems, where character-level many-to-many alignment is common and is difficult to model.

The CRF/SVM results using sub-syllabic units are listed in Table 1. The performance of the two approaches are similar: `SEG` is around .95 and `STR` is around .91. These results are acceptable, but there is still room for improvement. For comparison, Table 2 shows the results using syllables.[19] As the segments are different, the `SEG` of the two tables cannot be compared directly. However, `STR` in Table 2 is lower than in Table 1, with statistical significance at the $p < 1\%$ level. The main cause of this difference is the sparseness, which is evidenced by the percentage of unknown segments (`UNK`) in the tables. Specifically, there are 548 types of syllables in our data, while the sub-syllabic segmentation reduces the number of types to 136, which largely reduces the sparseness. Consequently, we conclude that the sub-syllabic units designed in this study provide an efficient interface for machine learning approaches to handle the Romanization task.

[15] http://taku910.github.io/crfpp/.
[16] http://www.phontron.com/kytea/.
[17] I.e., on the level in the bottom rank in Fig. 1, with no explicit alignment or unit boundaries between characters.
[18] `SEG` cannot be applied to the RNN approach as the alignment and segmentation are not explicit variables.
[19] I.e., the results in Tables 1 and 2 are based on the middle and upper-right parts in Fig. 1, respectively.

Table 1. CRF/SVM results on sub-syllabic units.

	2-fold	4-fold	8-fold
SEG	.946/.944	.948/.947	.947/.947
STR	.912/.907	.913/.911	.913/.910
UNK	0.13%	0.10%	0.09%

Table 2. CRF/SVM results on syllables.

	2-fold	4-fold	8-fold
SEG	.877/.882	.886/.888	.887/.889
STR	.878/.887	.888/.893	.890/.894
UNK	3.01%	2.48%	2.31%

7 Conclusion and Future Work

In this study, we focused on the Burmese name Romanization task, from the preparation of deliberately segmented and aligned data to experiments and discussion of statistical approaches. We are collecting more Burmese Romanization instances including the names of places and organizations. The annotated data and segmentation program have been released under a `CC BY-NC-SA` license to promote the research of NLP on Burmese.

Appendix

Figure 6 shows specific annotation instances for a further illustration and demonstration. The data are organized in a three-section format of

- original Burmese name,
- original Romanization, and
- aligned Burmese/Latin graphemes,

separated by `|||`.

The descriptions of specific instances are as follows.

I. An ordinary Romanization instance.
II. A Burmese name with a western expression (`Grace`) as a component. Generally, such western expressions are segmented according to the Burmese spellings. In this instance, `Grace` is segmented into `/G /@ /r /a /@ /ce`. Notice that we just apply the same `@` for the dummy vowel on Burmese side and for the silent placeholder on Latin side, which causes no confusion.
III. A Burmese name derived from Pali (`Wanna`),[20] where stacked consonants appear (`/n /n`). The stacked consonants are split and aligned to separate

[20] The Romanization instance is directly taken from the released data set. A more common Romanization of the Pali-derived name is `Wunna`.

I. ဒေါ်ဝင်းပပ ||| Daw Win Pa Pa ||| ဒ/D ော်/aw ဝ/W င်/in း/@ ပ/P @/a ပ/P @/a

II. ဂရေ့စ်နွမ်း ||| Grace Nuam ||| ဂ/G @/@ ရ/r ေ/a ့/@ စ်/ce န/N ွမ်/uam း/@

III. မောင်ဝဏ္ဏ ||| Maung Wanna ||| မ/M ောင်/aung ဝ/W @/a ဏ/n ္/@ ဏ/n @/a

IV. ရာဇသင်္ကြန် ||| Yar Za Thingyan ||| ရ/Y ာ/ar ဇ/Z @/a သ/Th င်/in ္/@ ြ/gy န်/an

V. နော်သီဂီထွန်း ||| Naw Theingi Htun ||| န/N ော်/aw သ/Th ိုင်/ein ့/@ ဂ/g ီ/i ထ/Ht ွန်/un း/@

VI. ဒေါ်သဉ္ဇာဝင်း ||| Daw Thinzar Win ||| ဒ/D ော်/aw သ/Th @/i ဉ/n ့/@ ဇ/z ာ/ar ဝ/W င်/in း/@

VII. ဒေါ်သဉ္ဇာဝင်း ||| Daw Thin Zar Win ||| ဒ/D ော်/aw သ/Th @/i ဉ/n ့/@ ဇ/Z ာ/ar ဝ/W င်/in း/@

Fig. 6. Specific annotation instances on Burmese name Romanization.

Latin letters. If no doubled Latin letters are used, the second Burmese character will be simply aligned to a silent placeholder `@`. The stacking operator is always aligned to `@`.

IV. A Burmese name with complex stacking, that the rhyme of the previous syllable (`/in`) is stacked with the following onset (`/gy`).

V. A Burmese name with more complex stacking, that part of the rhyme of the previous syllable (`/ein`) is stacked with the following onset (`/g`), which is taking a further vowel diacritic (`/i`). The instances IV. and V. illustrate the necessity on the segmentation of stacked characters.

VI. A Burmese name with stacked consonants, for which two syllables are kept as one word (`Thinzar`) in Romanization.

VII. A Burmese name with stacked consonants, for which two syllables are separated as two words (`Thin Zar`) in Romanization. Notice the Burmese names in instance VI. and VII. are identical. They are treated as two different Romanization instances due to the spellings in Romanization are different.

References

1. Banchs, R.E., Zhang, M., Duan, X., Li, H., Kumaran, A.: Report of NEWS 2015 machine transliteration shared task. In: Proceedings of NEWS, pp. 10–23 (2015)
2. Costa-Jussà, M.R.: Moses-based official baseline for NEWS 2016. In: Proceedings of NEWS, pp. 88–90 (2016)
3. Ding, C., Thu, Y.K., Utiyama, M., Finch, A., Sumita, E.: Empirical dependency-based head finalization for statistical Chinese-, English-, and French-to-Myanmar (Burmese) machine translation. In: Proceedings of IWSLT, pp. 184–191 (2014)
4. Ding, C., Thu, Y.K., Utiyama, M., Sumita, E.: Parsing Myanmar (Burmese) by using Japanese as a pivot. In: Proceedings of ICCA (Myanmar), pp. 158–162 (2016)
5. Ding, C., Thu, Y.K., Utiyama, M., Sumita, E.: Word segmentation for Burmese (Myanmar). ACM Trans. Asian Low Resour. Lang. Inf. Process. **15**(4), 22 (2016)
6. Finch, A., Liu, L., Wang, X., Sumita, E.: Neural network transduction models in transliteration generation. In: Proceedings of NEWS, pp. 61–66 (2015)
7. Finch, A., Liu, L., Wang, X., Sumita, E.: Target-bidirectional neural models for machine transliteration. In: Proceedings of NEWS, pp. 78–82 (2016)
8. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of ICML, pp. 282–289 (2001)
9. Liu, L., Finch, A., Utiyama, M., Sumita, E.: Agreement on target-bidirectional LSTMs for sequence-to-sequence learning. In: Proceedings of AAAI, pp. 2630–2637 (2016)
10. Naing, H.M.S., Hlaing, A.M., Pa, W.P., Hu, X., Thu, Y.K., Hori, C., Kawai, H.: A Myanmar large vocabulary continuous speech recognition system. In: Proceedings of APSIPA, pp. 320–327 (2015)
11. Neubig, G., Nakata, Y., Mori, S.: Pointwise prediction for robust, adaptable Japanese morphological analysis. In: Proceedings of ACL-HLT, pp. 529–533 (2011)
12. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Comput. Linguist. **29**(1), 19–51 (2003)
13. Okell, J.: A guide to the Romanization of Burmese (1971)

14. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of ACL, pp. 311–318 (2002)
15. Sha, F., Pereira, F.: Shallow parsing with conditional random fields. In: Proceedings of HLT-NAACT, pp. 134–141 (2003)
16. Thu, Y.K., Pa, W.P., Finch, A., Ni, J., Sumita, E., Hori, C.: The application of phrase based statistical machine translation techniques to Myanmar grapheme to phoneme conversion. In: Hasida, K., Purwarianti, A. (eds.) Computational Linguistics. CCIS, vol. 593, pp. 238–250. Springer, Singapore (2016). https://doi.org/10.1007/978-981-10-0515-2_17
17. Thu, Y.K., Pa, W.P., Ni, J., Shiga, Y., Finch, A., Hori, C., Kawai, H., Sumita, E.: HMM based Myanmar text to speech system. In: Proceedings of INTERSPEECH, pp. 2237–2241 (2015)