Towards Burmese (Myanmar) Morphological Analysis: Syllable-based Tokenization and Part-of-speech Tagging

CHENCHEN DING, ASTREC, National Institute of Information and Communications Technology, Japan
HNIN THU ZAR AYE, WIN PA PA, KHIN THANDAR NWET, and KHIN MAR SOE,
University of Computer Studies, Yangon, Myanmar
MASAO UTIYAMA and EIICHIRO SUMITA, ASTREC, National Institute of Information and Communications Technology, Japan

This article presents a comprehensive study on two primary tasks in Burmese (Myanmar) morphological analysis: tokenization and part-of-speech (POS) tagging. Twenty thousand Burmese sentences of newswire are annotated with two-layer tokenization and POS-tagging information, as one component of the Asian Language Treebank Project. The annotated corpus has been released under a CC BY-NC-SA license, and it is the largest open-access database of annotated Burmese when this manuscript was prepared in 2017. Detailed descriptions of the preparation, refinement, and features of the annotated corpus are provided in the first half of the article. Facilitated by the annotated corpus, experiment-based investigations are presented in the second half of the article, wherein the standard sequence-labeling approach of conditional random fields and a long short-term memory (LSTM)-based recurrent neural network (RNN) are applied and discussed. We obtained several general conclusions, covering the effect of joint tokenization and POS-tagging and importance of ensemble from the viewpoint of stabilizing the performance of LSTM-based RNN. This study provides a solid basis for further studies on Burmese processing.

CCS Concepts: • **Computing methodologies** → **Natural language processing**; *Phonology / morphology*;

Additional Key Words and Phrases: Burmese (Myanmar), annotated corpus, tokenization, POS-tagging, morphological analysis, CRF, LSTM-based RNN

ACM Reference format:
Chenchen Ding, Hnin Thu Zar Aye, Win Pa Pa, Khin Thandar Nwet, Khin Mar Soe, Masao Utiyama, and Eiichiro Sumita. 2019. Towards Burmese (Myanmar) Morphological Analysis: Syllable-based Tokenization and Part-of-speech Tagging. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 19, 1, Article 5 (May 2019), 34 pages.
https://doi.org/10.1145/3325885

5

This work Hnin Thu Zar Aye was done when the author was an intern student at ASTREC, National Institute of Information and Communications Technology, Japan, from June 2016 to May 2017.
Khin Thandar Nwet is currently at University of Information Technology, Myanmar.
Authors' addresses: C. Ding, M. Utiyama, and E. Sumita, ASTREC, National Institute of Information and Communications Technology, 3-5, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan; emails: {chenchen.ding, mutiyama, eiichiro.sumit}@nict.go.jp; H. T. Z. Aye, W. P. Pa, K. T. Nwet, and K. M. Soe, University of Computer Studies, Yangon, No.(4) Main Road, Shwe Pyi Thar Township, Yangon, Myanmar; emails: {hninthuzaraye, winpapa, khinthandarnwet, khinmarso}@ucsy.edu.mm.

2375-4699/2019/05-ART5 $15.00
https://doi.org/10.1145/3325885

1 INTRODUCTION

In linguistics, morphology studies the formation of meaningful units and the relation among them in specific languages. As for the engineering practice of natural language processing (NLP), automatic morphological analysis can be regarded as a general concept covering shallow processing related to basic meaningful units in textual data. Generally, automatic *identification* and *classification* of basic meaningful units within textual data are the two primary tasks in morphological analysis in NLP. The two tasks are usually referred to as *tokenization* and *part-of-speech (POS) tagging*, respectively.

The specific processing within morphological analysis is diverse and depends on different types of languages or, more formally, on linguistic typology. Consequently, most references pertaining to morphological analysis in NLP are language specific, because the features of languages largely affect engineering tasks. As for most inflected Indo-European languages, the processing includes related tasks such as stemming, lemmatization, and POS-tagging of words, where the core part revolves around the identification of stems and affixes. Also, this is true for many agglutinative languages with clear *word separators* in orthography, for example, Finnish, Turkish, and Korean [29], where the identification of various affixes is a main and heavy task. For languages without word separators in their scripts, further word segmentation or tokenization is required.[1] A typical example of an agglutinative language is Japanese. According to the definition in Neubig et al. [31], "*Japanese morphological analysis takes an unsegmented string of Japanese text as input, and outputs a string of morphemes annotated with parts of speech.*" The definition is given in a context of NLP engineering and the processing is briefly concluded as *cutting and tagging* textual strings. However, conjugated forms of numerous suffixes in Japanese should also be recovered in deeper analysis because of the agglutinativeness and the syllabic writing system [24]. A typical example of isolating languages is Chinese, where the term morphology analysis is less used, and the *cutting* processing is usually treated as a separate task called *Chinese word segmentation* [45], because there is no further recovery processing other than cutting when tokenizing an isolating language.

This study focuses on Burmese,[2] whose features can be thought of as a mixture of Chinese and Japanese. Morphologically, Burmese is highly analytic with no inflection of morphemes. Similar to Chinese, morphemes can be combined freely with no changes.[3] Syntactically, Burmese is typically head-final, where the functional dependent morphemes succeed the content independent morphemes and the verb constituent working as the root of a sentence always appears at the end of a sentence. Subordinative clauses are also placed before their modifying parts and before the main clause of a sentence. All these syntactic features are identical to Japanese.[4] Examples in Figures 1 and 2 show morphological and syntactic features of Burmese, respectively. Burmese word segmentation, which can be compared with Chinese word segmentation, has been investigated preliminarily in our previous work using in-house data [11]. In this study, we conduct a comprehensive study on the two primary tasks in Burmese morphological analysis, i.e., tokenization and POS-tagging of Burmese textual data by using the data prepared and released by us.

Based on the features of Burmese, this study can be regarded as a guide to *cutting and tagging* Burmese textual strings without any further insertion, substitution, or deletion during

[1]The terms "word segmentation" and "tokenization" are used interchangeably in this article.

[2]The language is referred to as Burmese or Myanmar in the literature. We use Burmese in this article, because English readers are more likely to be familiar with this name.

[3]Sandhi of consonant mutation from unvoiced to voiced may happen when morphemes are combined, but the phenomenon is not reflected in writing forms. A minor exception on contracted genitive case-marker for some nouns may affect the tone and spelling, as will be mentioned in Section 3.1.3.

[4]Some modifiers of nouns can be placed after the head noun they modify, which is an exception to the head-final restriction in Burmese. This will be mentioned in Section 3.1.3.

Fig. 1. Expression of "*minister of education*" in Burmese, which can be ultimately decomposed into seven morphemes as "*knowledge - affairs - officer - great - department - officer - great*." The process of the formation is shown from top to bottom. Notice the "*officer - great*," which means "*minister*," appears twice, and it takes part in the composition of "*ministry*" in the first time. All the morphemes have no inflection in the combination. The two morphemes for "*affairs*" and "*great*" in gray are largely grammaticalized while their original meanings of "*to compose*" and "*to be big*" are still preserved to a certain level.

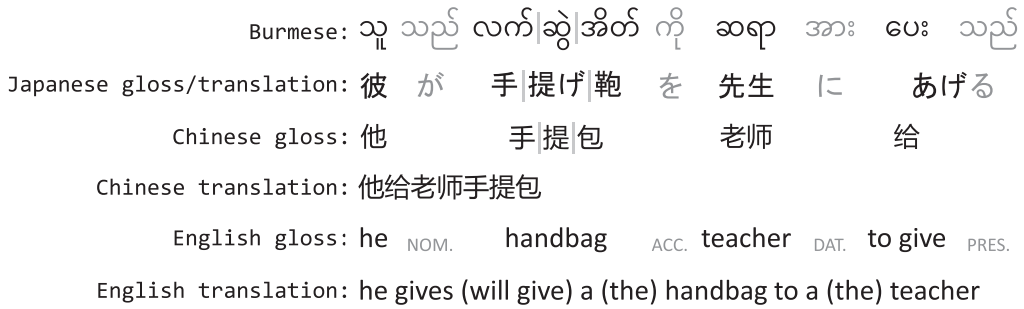

Fig. 2. A Burmese sentence and a comparison of Japanese, Chinese, and English. In the Burmese and Japanese sentences, the black parts are independent content words and the gray parts are dependent functional morphemes. In the English gloss, the corresponding functions of those dependent morphemes are noted by gray subscripts. The Japanese gloss of the Burmese sentence is grammatically correct translation, from which syntactical similarities between the two languages can be observed. Notice Burmese is more analytic than Japanese that the Burmese functional morphemes are all detachable but the Japanese verb ending is glued to the stem. There are more functional morphemes to annotate syntactic roles in the Burmese sentence than those in the Chinese sentence, where syntactic roles are afforded by fixed word order. The formation of the Burmese expression "*handbag*" is also shown by inserting vertical bars between morphemes, which is in the exact way as that in Chinese and Japanese (i.e., "*hand-to hang-bag*").

tokenization. This study contributes to both linguistics and NLP practice. Linguistically, we constructed an annotated Burmese corpus of around 20,000 sentences with two-layer tokenization and POS-tagging to provide morphological information. The two-layer scheme is originally a solution from annotation practice of a large-scale Japanese corpus [32] regarding to problematic cases in tokenization. We design the annotation scheme to cover basic and important linguistic phenomena in Burmese. Seven rounds of cross-checking on the 20,000 Burmese sentences were conducted to achieve consistent and precise annotation. This is thus the best-prepared morphologically annotated Burmese corpus in terms of quality and quantity as of the writing of this article in 2017. The

corpus has been released under a CC BY-NC-SA license for the research community,[5] as one component of the Asian Language Treebank (ALT) Project.[6] The guidelines of the annotation are also available to the public to provide more details for users of the corpus.[7] For NLP practice, we have experimented with two mainstream engineering approaches in tokenization and POS-tagging of Burmese, namely, a classical and standard sequence-labeling approach of conditional random fields (CRFs) [25], and a state-of-the-art approach of long short-term memory (LSTM)-based [19] recurrent neural network (RNN) [17]. Based on the experimental results, we obtained several general conclusions, covering the effect of joint tokenization and POS-tagging, and importance of ensemble in stabilizing the performance of LSTM-based RNN. This study, thus, provides a solid basis for further studies on Burmese processing, such as syntactic parsing and machine translation.

The remainder of the article is organized as follows. In Section 2, we discuss related work on general approaches to morphological analysis in NLP, and previous work on the processing of Burmese. In Section 3, a detailed description of the annotated corpus is provided. Because the final data and guidelines have been released, the motivation and the design of the overall annotation scheme, with several important issues pertaining to data annotation and refinement, are presented in this section. Experiments related to CRFs and LSTM-based RNNs are presented in Sections 4 and 5, respectively, where various explorations and comparisons are presented. Section 6 presents a discussion of the experimental results. Section 7 presents our concluding remarks and lists our future work on Burmese and other Southeast Asian languages.

2 RELATED WORK

Shallow processing tasks in NLP, such as word segmentation, POS-tagging, and chunking, can generally be modeled as classification tasks to label tokens, or structured prediction tasks considering sequential features of textual data. A classical and standard approach to such tasks is CRF, i.e., a probabilistic graphical model that can be applied easily to sequential labeling tasks. Typical works on the application of CRFs to morphological analysis are as follows: Kudo et al. [24] to Japanese morphology analysis, Zhao et al. [45] to Chinese word segmentation, Na [29] to Korean morphology analysis, and our note on Burmese word segmentation [11]. Also, non-structured approaches, such as classifiers of support vector machine (SVM), are widely studied and applied in practice. Typical SVM-based works are as follows: Japanese morphology analysis by Kudo and Matsumoto [22][8] and Neubig et al. [31]. Such approaches may apply a dynamic programming method to integrate the surrounding information on output tags to achieve a similar effect of structured learning [22] or simply adopt a pure point-wise approach [31] to achieve fast processing by light-weight models. However, Stratos and Collins [38] illustrated that well-programmed point-wise approaches can actually yield results comparable to those achieved using standard CRFs for POS-tagging of several Indo-European languages. A similar conclusion is also reached in Ding et al. [11], in that the performance difference between CRFs and point-wise SVM is not considerable. It can be considered that the capacity of a general supervised machine learning framework is adequate for shallow processing tasks in NLP and that the difference caused by structured learning is not significant, so long as the classifier is well trained.

In addition to the classical feature-based approaches, neural network (NN)-based approaches have been studied overwhelmingly in the context of NLP in recent years. An early comprehensive

[5] http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/my-nova-170405.zip.
[6] http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/.
[7] Basic guidelines: http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/Myanmar-annotation-guideline.pdf; and Supplementary instructions: http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/Myanmar-annotation-guideline-supplemantary.pdf.
[8] A more detailed version in Japanese is Kudo and Matsumoto [23].

work on NLP is Collobert et al. [6], where various classification and sequence labeling tasks in NLP are processed in a unified manner by NNs. Generally, an NN-based approach involves pure end-to-end processing, where the relation between input and output is modeled directly by connected nonlinear units. The crucial issues of NN-based approaches are as follows: (1) network topology, i.e., how to connect different basic units, and (2) the composition of basic units, i.e., how to conduct nonlinear transformation. Two typical NN structures are convolutional neural network (CNN) and RNN. As for NLP tasks, RNN is more popular than CNN, because it better fits the sequential features in textual data [28, 39]. The nonlinear units can be simple s-shape nonlinear functions, for example, sigmoid function or hyperbolic tangent, or a more complicated and powerful block, such as the LSTM unit, which has become a common component of RNNs. LSTM-based RNN, thus, has been used as a standard state-of-the-art approach in various NLP tasks. As for the related works on morphology analysis, typical studies include Chen et al. [4] and Ma and Hovy [26] on LSTM-based approaches for Chinese word segmentation.

In terms of research on Burmese processing, early studies are largely based on decisive approaches (e.g., automaton or dictionary-based maximum matching), or applying simple corpus-based statistical metrics (e.g., bag of words and N-gram counting or mutual information), due to the lack of well prepared resources. Typical early works are Hla Hla Htay et al. [18], Aye Myat Mon et al. [1], and Thet Thet Zin et al. [41]. Recently, there are two comprehensive data-driven works: our previous note on word segmentation of Ding et al. [11] and the work of Khin War War Htike et al. [20] on POS-tagging, both of which are based on relatively large annotated data prepared by the authors. In Ding et al. [11], various word segmentation approaches were compared using a tokenized Burmese dataset with over 60,000 sentences. The study illustrated that data-driven supervised approaches outperform rule-based matching and unsupervised approaches. However, the differences among different supervised approaches are insignificant. The drawbacks of the study are as follows: (1) no detailed comparison in feature engineering, (2) annotated data show relative inconsistency among annotators, and (3) in-house data cannot be shared by the research community. The work of Khin War War Htike et al. [20] compared various POS-tagging approaches using a POS-tagged Burmese dataset comprising 11,000 sentences.[9] However, a limitation of this study is that all experiments were conducted based on the tokenization of the specific dataset. That is, various models were trained and tested using manually tokenized data. Given that "*words*," i.e., tokens in processing, are not natural units in Burmese texts, this study is not oriented for a practical setting, because all details pertaining to tokenization are omitted by considering there is a perfect tokenizer.

This study covers all ranges of the two previous studies. Essentially, this study is a natural extension of the previous note of Ding et al. [11] in terms of open-access data and further experiment-based investigation. We have released a high-quality Burmese corpus comprising around 20,000 sentences and experimented with representative approaches of CRFs and LSTM-based RNNs with various comparisons. Consequently, this study provides reliable conclusions and a solid benchmark of the numerical results related to Burmese tokenization and POS-tagging tasks.[10]

[9] https://github.com/ye-kyaw-thu/myPOS.

[10] The data of Khin War War Htike et al. [20] were draft released when this study conducted. The released data are not identical to the data used in their publication and the data have been further updated after we conducted the experiments reported in this article. Therefore, the numerical results obtained using these data are thus not strictly comparable among the different references thus far (2017). We used the dataset in this study in an auxiliary way to confirm the conclusions we obtained hold across different datasets. The data used in this study can be accessed at https://github.com/chenchen-ding/mycicling.

3 ANNOTATED BURMESE CORPUS

3.1 Asian Language Treebank (ALT) Burmese Corpus

3.1.1 Overview. An overview of the ALT project and international collaboration can be found in the report by Riza et al. [36]. Briefly, 20,000 English sentences collected from *Wikinews* were translated manually into different Asian languages as the raw data.[11] Further annotations were then conducted for each language, including tokenization, POS-tagging, phrase-structured tree-building, and token alignment with the original English sentences. The Burmese language is the first Southeast Asian language processed in the ALT project. As this study focuses on morphological analysis of Burmese, in this article, the introduction of ALT Burmese data is restrained on the the morphological annotation tasks of tokenization and POS-tagging. The tree-building task and syntactic parsing of Burmese is our work in recent future, which will be established on the basement of this study.

We designed a unified framework called nova [9] for annotating low-resourced but highly analytic languages from scratch, based on the following three motivations.

- a compact tagset for cross-lingual word classes, e.g., nouns and verbs
- extensibility to language-specific word classes, e.g., pronouns, articles, and so on
- flexibility to the concept of words, i.e., can tolerate and represent the ambiguities

The scheme of nova provides four basic tags, namely, n, v, a, and o, to represent fundamental word classes of nouns, verbs, adjectives, and other modification tokens, with three additional auxiliary tags to represent numbers (1), punctuations marks (.), and tokens with weak syntactic roles (+). Basic tags can be further modified by a - mark, where the *functionality* is addressed.[12] In addition to the basic, auxiliary, and modified tags, a pair of brackets ([and]) is further applied to provide two-layer annotation to adapt to the ambiguities in tokenization, as well as to annotate larger syntactic constituents that can be applied as an integrated unit for further syntactic parsing.

An example of an annotated Burmese sentence is shown in Figure 3. Because the guidelines for annotating Burmese data have been released, we do not provide detailed instructions here but present an introduction of the overall process and the features of the annotated Burmese data.

3.1.2 Process of Annotation. Preparation of the 20,000 raw Burmese sentences involved around 150 translators, including professional translators and faculty members from *University of Computer Studies, Yangon* (UCSY). At the translation stage, the *Unicode* encoding standard[13] is adapted instead of Myanmar local *Zawgyi* typeface.[14] Then 25 annotators, who are students and faculty members from UCSY, performed a preliminary annotation. As most annotators were inexperienced, the process was mentored by experienced native-speaker researchers (mainly by the third author of this article). At this stage, overall principles based on rough grammatical analysis were carried out, using a traditional POS tagset on Burmese, basically according to the *Myanmar-English Dictionary* [7] and the *Myanmar Grammar* [8] edited by the *Myanmar Language Commission*. However, relatively serious inconsistency in the preliminary annotation, regarding to morphological

[11] Specifically, they are Burmese (Myanmar), Indonesian, Japanese, Khmer, Laotian, Malay, Tagalog (Filipino), Thai, and Vietnamese. All of the languages are low-resourced except Japanese.

[12] In annotating Burmese, the o- tag is used to annotate a large range of functional tokens, which afford most syntactic information in Burmese; the n- tag is used for various pronouns, including personal, demonstrative, interrogative, and numeral ones; the a- tag is used for determiners, which are most derived from n- and as direct modifiers for n and or n- tokens. The v- tag is not used. A portion of verb-derived particles are tagged as o-, as it is non-trivial to measure the extent of various phenomena of grammalization.

[13] https://www.unicode.org/charts/PDF/U1000.pdf.

[14] https://code.google.com/archive/p/zawgyi/downloads.

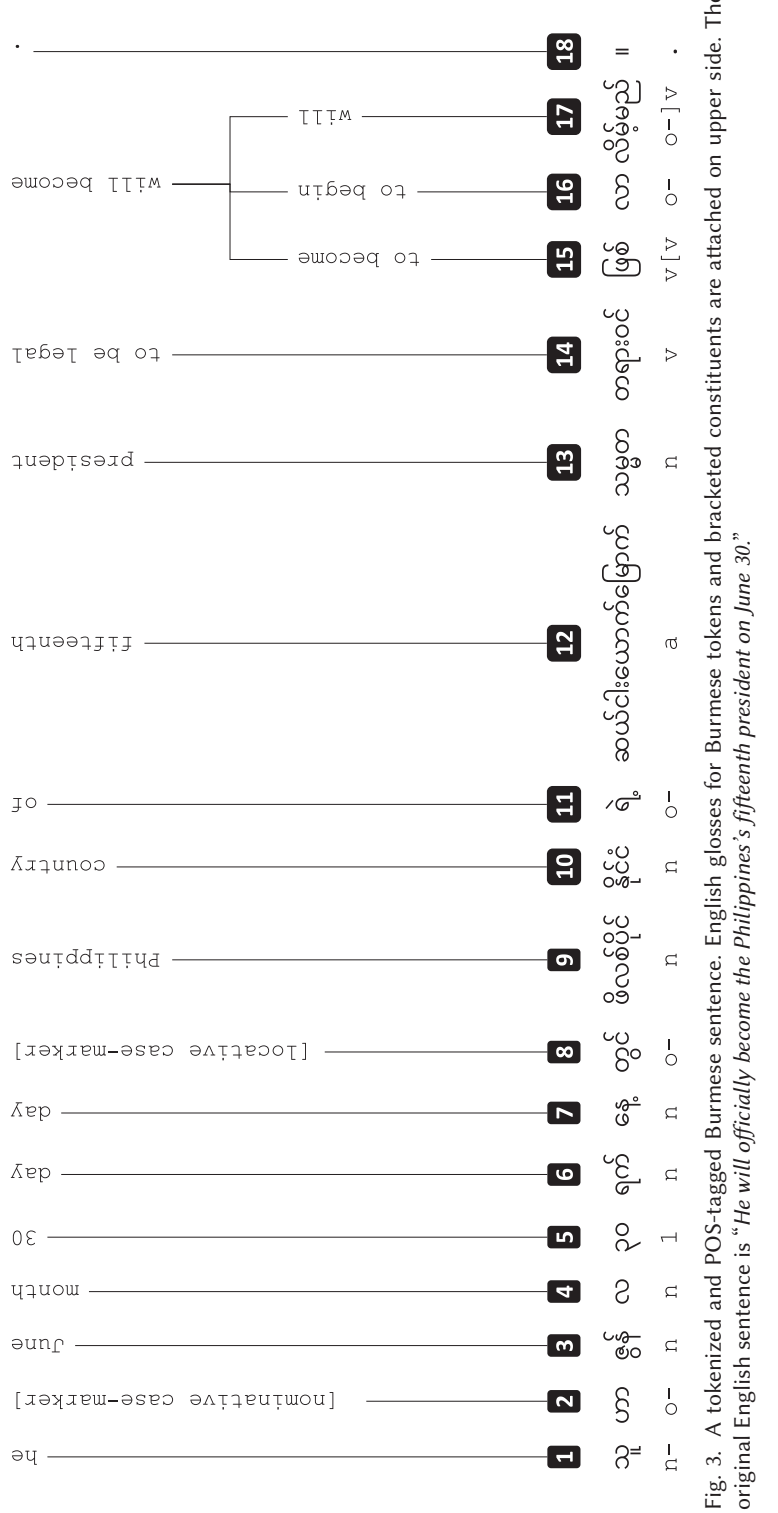

Fig. 3. A tokenized and POS-tagged Burmese sentence. English glosses for Burmese tokens and bracketed constituents are attached on upper side. The original English sentence is "*He will officially become the Philippines's fifteenth president on June 30.*"

Table 1. Statistics on Eight-fold Cross-validation of Annotated ALT Burmese Data in Different Versions

data version	pattern error		#tag error	#syllable	tag error rate
	type	number			
original	54,059	93,269	138,630	1,170,587	11.84%
16-10-31	33,048	48,672	71,939	1,171,548	6.14%
16-11-14	31,825	42,230	62,599	1,171,577	5.34%
17-01-01	28,580	37,150	52,426	1,171,107	4.48%
17-01-15	28,144	36,763	51,315	1,171,218	4.38%
17-01-26	27,598	36,177	49,980	1,170,980	4.27%
17-03-09	27,332	35,996	50,449	1,170,980	4.31%
17-04-05	26,418	35,130	48,869	1,170,922	4.17%

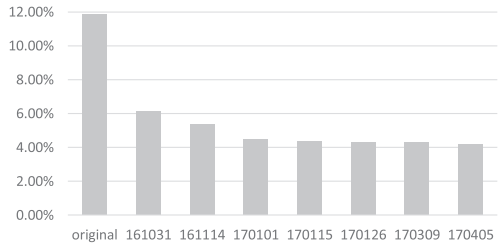

Fig. 4. Tag error rate in Table 1.

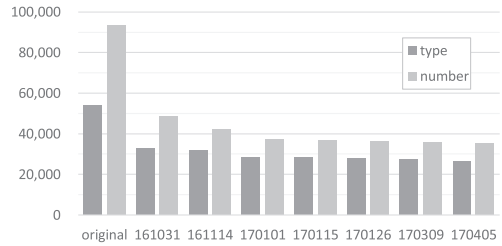

Fig. 5. Type/number of error patterns in Table 1.

segmentation and classification,[15] prevented from further maintenance and refinement of the annotation. To relieve the problem and to provide basic and consistent information on morphological level, the preliminarily annotated Burmese data were converted into the nova scheme, where the ambiguous and inconsistent cases were kept within the brackets to the extent possible. An eight-fold cross-validation using CRF was then applied on the entire dataset by tri-gram features of syllables [11] for further refinement.

Based on the results of cross-validation, the annotation of the data was modified and improved in a systematic way. Specifically, the errors in cross-validation were extracted, ranked by frequency, and examined case by case. As these errors may be caused due to the inconsistency of annotation or the limitation of the automatic cross-validation, we did not only focus on specific inconsistent patterns but also determined the linguistic phenomena that were annotated inconsistently, and modified them to be consistent. Automatic cross-validation and manual refinement were applied repeatedly to improve the quality of annotation. The annotation guidelines were also updated interactively along with the process of refinement. Table 1 lists statistical information about the data of different versions under the refinement processing, where the original refers to the preliminarily annotated data and the latest 17-04-05 data comprise the final released version.

Specifically, **#tag error** in Table 1 is the count of syllables with wrong tags in the cross-validation, and **#syllable** is the total number of syllables. The **tag error rate** (graphed in Figure 4) is thus the quotient of **#tag error** divided by **#syllable**. The **type** and **number** of **pattern error**s (graphed in Figure 5) are counted by the maximum-length-matching of the wrongly tagged

[15]For example, in tokenization of complex nominal and verbal expressions, and in tagging various functional morphemes. The unrefined annotated data at this stage can be referred to at http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/Myanmar-ALT-20170110.zip.

syllable sequences. In versions of 16-10-31 to 17-01-26, the data are annotated using four basic tags (n, v, a, and o) and brackets ([,]), the details of which are available in the basic guidelines. The tags in versions 17-03-09 and 17-04-05 are modified further by attaching functional marks (- and /o-), which can be referred to in the supplementary instructions of annotation. It can be observed that both the tag error rate and the pattern errors converge along the refinement. Notice the tag error rate increased slightly from version 17-01-26 to version 17-03-09, because the number of tag types increased upon the modification, while the number of pattern errors decreased stably, indicating the data were always improved in terms of consistency.

In Table 1, it can be observed that the number of syllables changes slightly across different versions. This is because the spellings were modified and normalized along with annotation refinement. There are two main normalizations as follows.

- The order of the creaky tone-marker aukmyit (U+1037) and the inherent vowel-depressor virama (U+103A) is arranged in "aukmyit virama" in coding nasal-ended creaky-toned syllables, in accordance with the processing in Ding et al. [10].[16] The order of the two diacritics is quite inconsistent in daily typing, while the order we adopt is considered as the standard order and is supported by different fonts.
- The Burmese letter wa (U+101D) and the Burmese digit zero (U+1040) are both o-shaped character and are extremely similar to each other (if not identical in some fonts).[17] The two characters are used interchangeably in casual typing. Therefore, we thus paid specific attention to avoid their misuse case by case.

Compared with the work of Ding et al. [11] where no cross-checking was applied to control the annotation quality, we paid considerable efforts to improve the quality of the annotated corpus in this study. The consistency in Burmese spellings and the usage of tags were improved exhaustively under the iterative automatic-manual refinement. As shown in Figures 4 and 5, the error rate and patten errors reached a stable plateau where further improvement will not be obvious under the refinement framework we applied. Compared with the English *Penn Treebank*, where the inconsistency in POS-tagging is around 3%–5% [27, 40], we consider the quality of released version of the ALT Burmese data is acceptable for NLP research.

3.1.3 Features and Statistics. Based on the design and the practice of annotation processing, the tokenized and POS-tagged Burmese data have specific features in addition to the quality and quantity. We conclude them as follows.

- A two-layer tokenization and POS-tagging annotated by brackets covers a portion of the ambiguities in tokenization and identifies the composition of specific constituents. Typical examples are shown in Figure 3, where tokens 15 16 17 compose a multiply suffixed verbal constituent. The two-layer annotation addresses a large range of linguistic phenomena in Burmese, for instance, derivation, compounds, heavily agglutinative constituents, reversed nominal-attributive constituents, and number-counter constituents.
- The tags contain analytic information. Four basic tags are used for sketchy annotation, and further modification of functionality (-) and contraction (/o-) are added to the basic tags for providing more detailed information. As for the example in Figure 3, o- tags are used

[16]Notice the middle part of token 17 in Figure 3. The lower tiny circle is aukmyit, and the upper arc is the virama.

[17]Notice the standalone full circles in tokens 5 and 14 in Figure 3. The circle in 5 is the Burmese digit zero and in 14 the Burmese letter wa.

Table 2. Statistics on the Released ALT Burmese Data
(the version of 17-04-05 in Table 1)

dataset	#syllable	#token		#sentence
		short	long	
training	1,054,829	664,174	498,227	17,965
development	57,607	36,133	27,081	993
test	58,486	36,830	27,740	1,007
total	1,170,922	737,137	553,048	19,965
average syllable(s)	1	1.59	2.12	58.65

to annotate various affixes and particles.[18] The token **11** is a genitive case-marker that can be contracted into a creaky tone, where the /o- will be attached to an n or n- tag to annotate this contracted genitive case-marker.

The annotated Burmese corpus, thus, contains useful information, which is feasible and flexible for various downstream NLP practical applications. The corpus also provided a refined and stable platform for academic research on investigating techniques for Burmese morphology analysis. For this purpose, the entire corpus was split into three datasets for training, development, and test for experiments and comparison. The statistics pertaining to the Burmese corpus are listed in Table 2,[19] where **short token** is the number of finally segmented tokens and **long token** is countered by considering bracketed tokens as one token. As for the example in Figure 3, there are 18 short tokens and 16 long tokens.

Compared with the in-house data used in Ding et al. [11], the annotated Burmese corpus in this study has fewer sentences but it contains more syllables and is segmented into smaller tokens. Comparing Table 2 here and Table 1 in Ding et al. [11], we find there are more than one million ($\mathcal{M}$) syllables in this ALT dataset but only fewer than 0.8 $\mathcal{M}$ syllables in the in-house dataset. The average sentence length thus differs largely between the two datasets, and there are only 12.67 syllables per sentence in the in-house dataset. The difference is caused mainly by the genre of the textual data in the corpora. The ALT data are composed of news articles, where formal and long sentences are common, while, as stated in Ding et al. [11], the in-house data are restricted to travel expressions, which are generally simple in syntax and vocabulary. It can also be deduced that there are 1.63 syllables per token (word) on average in the previous in-house dataset. Therefore, the Burmese data used in this study are tokenized more finely, although the topic and field are more complicated.

3.2 CICLING Burmese Corpus

In this study, we report further experimental results on the data used in Khin War War Htike et al. [20], which will be called CICLING data. The statistics on the version we used herein are presented in Table 3. Because there is no explicit division of the dataset, we selected the test data by each eleven sentences from the entire dataset, and selected the development data by each ten sentence in a same way from the remainder.

[18] The basic o tag is used to annotate a general content modifier (e.g., adverb).

[19] The three datasets are formed by dividing the corpus comprising articles from the original English *Wikinews*. The lists of original URLs are available at http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/index.html. Notice the total number of sentences listed in Table 2 is slightly smaller than that in the raw data on the linked page. This is because wrongly translated or segmented sentences were excluded from the annotation process.

Table 3. Statistics on the CICLING Burmese Data Used in This Study

dataset	#syllable	#token		#sentence
		short	long	
training	303,588	194,024	-	9,000
development	34,675	22,172	-	1,000
test	33,335	21,315	-	1,000
total	371,598	237,511	215,931	11,000
average syllable(s)	1	1.56	1.72	33.78

The annotation structure of these data is simpler than that of ALT data, where tokenized Burmese textual data are annotated using 15 different tags. There was one more tag for a negative particle in the original publication but it was removed in later version, because this tag was used exclusively for only one pre-positional negative particle. However, the temporary tagset used in the CICLING data can be designed and applied more accurately. For example, there are tags for abbreviation (abb), foreign words (fw), and text numbers (tn), which are not decided by syntactic roles, but by surface spellings. Although the two tags of abb and fw are mostly used for nominal tokens in the data, according to the examples in the instructions, abb is also used for adverbs and fw for Arabic numbers. The differences between particle (part) and post-positional marker (ppm) are also not obvious. It seems those ppm-tagged tokens are only restricted to a portion of post-positional case-markers, and the other post-positioned functional tokens are all classified to be part.

Some compounds are also annotated in the corpus by concatenating several tokens. Compared to the ALT data, the annotation is not systematic and complete, and there is no further POS annotation for the larger concatenated tokens.[20] In Table 3, the basic and concatenated units are represented as **short** and **long token**, respectively, for the sake of a comparison of the ALT data. The sizes (number of syllables) of short tokens are nearly the same in the two corpora, because the tokenization principles are not very different from each other. In the CICLING data, the size of long tokens is not very different from that of short tokens, because only a few concatenations are annotated. In the ALT data, the average length of long tokens is around 133% that of short tokens, while in the CICLING data, it is only 110%. Compared with the ALT data, the CICLING data have weak and incomplete two-layer annotation. Therefore, we conducted experiments using short tokens of the CICLING data in this study, because long tokens are insignificantly different from short tokens, and there is a lack of POS information about long tokens.

3.3 On Annotation System and Future Development

We have introduced the annotated Burmese corpora of ALT and CICLING data. As a conclusion of the first half of this article, here we provide a general discussion on annotation system and the future development of Burmese corpus construction.

In Table 4, we present a comparison of the universal POS tags [34] designed for cross-lingual processing, with the POS tags used in the ALT and CICLING data, respectively. There are twelve different tags in the **original** version of universal POS tags. An **extended** version with several subdivided and added tags is used in the *Universal Dependencies* project.[21] The seven tags marked

[20] Most cases are nominal expressions from our observations. A token annotated as n|n usually means that it is a compound noun, composed of two nouns. The composition is annotated by the vertical bar here.
[21] http://universaldependencies.org/.

Table 4. Comparison of Universal POS Tags with POS Tags Used in the CICLING and ALT Burmese Data

universal POS tags				
original	**extended**	**CICLING**	**ALT**	**note**
ADJ*		adj	a	
ADP		ppm	o-	ppm for *post-positional marker*, mainly case-markers
ADV*		adv	o	
CONJ	CCONJ SCONJ	conj	o-	
DET			a-	a- used for a couple of determiners
NOUN*		noun	n	n/o- for nouns with contracted genitive marker
	PROPN			
NUM*		num, tn	1	
PART		part	o-	
	AUX			
PRON*		pron	n-	n-/o- for pronouns with contracted genitive marker
PUNCT*		punc	.	
VERB*		verb	v	
X			+	
	SYM	sb		
	INTJ	intj		
		abb		abb for *abbreviation*, not a grammatical category
		fw		fw for *foreign words*, not a grammatical category

with * in the original version of universal POS tags have their correspondences both in ALT and CICLING annotations. As an obvious feature of the ALT annotation, the adposition (ADP, in the case of Burmese, postposition), conjunction (CONJ), and particle (PART) in universal POS tags are not distinguished but covered by an o- tag in the nova scheme. From the original definition, the o- tagged tokens can be interpreted as functional tokens of closed word classes. Generally, the grammatical functions in Burmese are mainly afforded by a set of post-positional morphemes [33], and their syntactic roles can be largely identified by their own lexicon form with the type of content morphemes preceding them. Hence, the merging of tags for various post-positional functional morphemes will not lead to serious confusion, if lexical information of these functional morphemes is applied in NLP tasks. As the nova scheme is designed for a gradual coarse-to-fine annotation from scratch, the temporary annotated ALT data have achieved to provide basic grammatical information, based on which more detailed grammatical information can be further identified. The annotation system in CICLING is more like the extended version of universal POS tags. The determiner (DET) is not available is the CICLING data. As it is quite a small word class, this is not a serious defect. Notice even the CICLING data apply more detailed tags on functional morphemes, there is no distinction between coordinative conjunction (CCONJ) and subordinative conjunction (SCONJ), as well as between particle (PART) and auxiliary verb (AUX). As mentioned, such grammatical roles are realized by various post-positional functional morphemes, that an excessively detailed tagging scheme provides less increment in information but more confusion and inconsistence in annotation.

Generally, the data size, the informativeness of an annotation scheme, and the quality of manual annotation are difficult to achieve at the same time. Considering the Burmese language is still a low-resourced language, with limited resource, research, and researchers, the ALT data focus on

the size and the quality of annotation. That is, the ALT data prepare a foundation by considerable data size and good quality of annotation, with basic necessary grammatical information for NLP practice. As a natural extension of temporary work, we consider the following issues for future development on Burmese morphological annotated data.

- To apply the universal POS tags for cross-lingual processing, and to develop more informative Burmese-specific annotations. This line of work requires more detailed and systematic analysis in terms of Burmese grammatical features, which can be established on the available ALT and CICLING data.
- To build a large-scale, well-organized, and constantly updated neologism dictionary. Such a work for Japanese [43] supports the Japanese morphology analysis in processing everyday textual data. Given the temporary prepared ALT and CICLING data, such a large dictionary for Burmese may bring more direct gains in NLP practice than only increasing the number of annotated sentences.

Here, we finish the first half of this article. Based on the aforementioned ALT and CICLING data, engineering approaches and experiments for Burmese tokenization and POS-tagging tasks will be presented in the second half of the article.

4 BURMESE TOKENIZATION AND POS-TAGGING BY CRF

4.1 Feature, Tag, and Tool

Generally, the CRF model can be formulated within the framework of the maximum-entropy principle, as in Equation (1):

$$p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\lambda}) = \frac{\exp(\sum_j \lambda_j f_j(\boldsymbol{y},\boldsymbol{x}))}{\sum_{\boldsymbol{y}} \exp(\sum_j \lambda_j f_j(\boldsymbol{y},\boldsymbol{x}))}. \tag{1}$$

For a sequential labeling task, in Equation (1), $\boldsymbol{x}$ represents a sequence of input tokens $x_0^i = x_0, x_1, \ldots, x_i$, and $\boldsymbol{y}$ represents a sequence of output labels $y_0^i = y_0, y_1, \ldots, y_i$, for the corresponding tokens with the same index. f_j is a feature function and λ_j is the corresponding feature weight. Thus $\boldsymbol{\lambda}$, the set of λ_j, is the parameter of the model. For a given parameter $\boldsymbol{\lambda}$ and a token sequence $\boldsymbol{x}$, the most likely labeling sequence is $\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\lambda})$. To obtain a sound model parameter $\boldsymbol{\lambda}$, it should be tuned using a set of training data $\{\boldsymbol{x}^0,\boldsymbol{y}^0\},\{\boldsymbol{x}^1,\boldsymbol{y}^1\},\ldots,\{\boldsymbol{x}^k,\boldsymbol{y}^k\}$. Generally, $\boldsymbol{\lambda}$ is optimized by maximizing the following log-likelihood of Equation (1) using training instances:

$$\mathcal{L}(\boldsymbol{\lambda}) = \sum_k \left\{ \left(\sum_j \lambda_j f_j(\boldsymbol{y}^k,\boldsymbol{x}^k) \right) - \log \sum_{\boldsymbol{y}^k} \exp\left(\sum_j \lambda_j f_j(\boldsymbol{y}^k,\boldsymbol{x}^k) \right) \right\}. \tag{2}$$

Here, we only provide an overview of the CRF framework without mentioning the details of optimization and implementation, because they are already parts of the mature framework of the algorithm, and off-the-shelf tools are available for practice. In this study, we focus on feature selection and tagset for labeling, which is the most practical issue in a morphological analysis task. The notation of $f(\boldsymbol{y},\boldsymbol{x})$ in Equation (1) is in a rather generalized form, and it can be decomposed further into specific tasks. In the interface of tokenization and POS-tagging tasks in NLP, the features used are usually binary on a sliding window for each position of a token-label pair, with local contextual information from previous and succeeding tokens. Therefore, Equation (1) can be transformed into Equation (3), where x_m^n offers a context window for the label y_k, and δ_j is the Kronecker delta, where the value of a feature is 1 if and only if x_m^n and y_k match exactly; otherwise it is 0. Notice that Equation (3) yields the probability of one training instance, while δ_j refers to cross-instances, that is, the co-occurrence of specific x_m^n and y_k can appear in different instances.

The weight λ_j can thus be intuitively interpreted as how "important" the co-occurrence of x_m^n and y_k should be regarded over all given training instances. Although the features are restricted to local windows, the normalization in Equation (3) is over the entire sequence of labels (i.e., the summation over y_0^i in denominator), so a $\hat{y}_0^i = \arg\max_{y_0^i} p(y_0^i|x_0^i, \boldsymbol{\lambda})$ is still searched for under a global optimum:

$$p\left(y_0^i|x_0^i, \boldsymbol{\lambda}\right) = \frac{\exp(\sum_{k=0}^{i} \sum_j \lambda_j \delta_j(y_k, x_m^n))}{\sum_{y_0^i} \exp(\sum_{k=0}^{i} \sum_j \lambda_j \delta_j(y_k, x_m^n))} \quad (0 \le m \le k \le n \le i). \tag{3}$$

Consequently, there are two basic issues with CRFs that we should experimentally investigate, in terms of y_k and x_m^n, respectively.

- A feasible tagset for y_k. The issue is trivial in a pure POS-tagging task, as in Khin War War Htike et al. [20], where the POS tagset is used directly. In the tokenization interface, however, the tokens' boundaries should be identified, and thus, variants of notation differ when addressing the beginning or the end of a token. The number of different tag types (i.e., the tagset size) should also be considered in practice. As a larger tagset size results in a larger space of $\boldsymbol{y}$ in the normalization term of Equations (1) and (2), which requires more calculations in model training and decoding.
- A proper window size, that is, the magnitude of $n - m$ in x_m^n, is required. Generally, if we use a large window size, more contextual information can be modeled, but it will cause feature sparseness, where superfluous patterns of x_m^n and y_k, that is, numerous δ_j, will be collected from the training instances, which makes the parameter training slow and inadequate.

We adopt the Burmese syllable as the basic unit in the morphological analysis, which is reasonable in terms of both NLP engineering and linguistic features of Burmese. Syllables can be identified decisively by rules [2, 42] and considered as the atoms of Burmese morphemes, because no boundary of morphemes locates within a syllable. We applied the straightforward way in Ding et al. [11] to conduct the syllable segmentation, where all the dependent characters[22] and character combinations, i.e., "*asat-ed*" characters,[23] are attached to the independent characters they modify.[24] This process guarantees the primary requirement, to prevent a statistical model from generating morphological boundaries at an impossible location. The follows are two special issues related to Burmese orthography.

- The contracted genitive case-marker that becomes a creaky tone denoted by a diacritic, is annotated with specific tags attached by `/o-` (i.e., `n/o-` or `n-/o-`) in `nova` scheme. Therefore, the only exceptional sub-syllabic analysis is not required anymore under the annotation in the ALT data.
- The staked consonant letters (e.g., Figure 3 in Ding et al. [11]) are kept together,[25] in accordance to the overall prevention of improper boundary insertion in automatic processing. This kind of special spelling mainly appears in traditional loan words from Sanskrit and Pali, and modern words borrowed from western languages. The pattern and frequency of stacked letters are restricted, and hence this will not introduce much extra cost in processing.

On the contrary, if units smaller than syllables are adopted in statistical models, then extra parameters are required to describe the probabilities over the combination of these sub-syllabic units,

[22] *Unicode* character from `102B` to `1038` (vowel and tone diacritics) and from `103B` to `103E` (dependent consonants).
[23] characters modified by *Unicode* character `103A`, the devowelizer.
[24] *Unicode* characters from `1000` to `102A` and from `103F` to `104F`.
[25] *Unicode* character `1039` is the stack operator, which glues its preceding and succeeding syllables.

REL. IDX.:	-5	-4	-3	-2	-1	0	1	2	3	4	5
SYLLABLE:	သ	မ္မ	တ	တ	ား	ဝင်	ဖြစ်	လာ	လိမ့်	မည်	။
SMALL-TOK:	B	I	E	B	I	E	S	S	B	E	S
LARGE-TOK:	B	I	E	B	I	E	B	I	I	E	S
SMALL-POS:	B-n	I-n	E-n	B-v	I-v	E-v	S-v	S-o-	B-o-	E-o-	S-.
LARGE-POS:	B-n	I-n	E-n	B-v	I-v	E-v	B-v	I-v	I-v	E-v	S-.

Fig. 6. IBES tagging scheme for tokens 13 14 15 16 17 18 in Figure 3. By changing B to I and S to E, it turns into an IE scheme; by changing E to I and S to B, it turns into an IB scheme. An instance of stacked consonant letters appears at position -4, which is kept together.

where morpheme boundaries never locate. A potential merit of using sub-syllabic units is that it can reduce the types of units in processing. However, the types of Burmese syllables in writing are only at a magnitude of 10^3, which is a reasonable number for statistical approaches to model.[26] As a matter of fact, the syllable-based processing for Burmese introduces a decisive pre-processing, where *a prior* knowledge of linguistic features of Burmese is integrated and a neat interface is provided to statistical models.

Based on the syllables, we investigate different tagging schemes and feature templates. For shallow processing tasks, various tagging schemes have been proposed [13, 35, 44], among which an IBES notation has been adopted as a default scheme in temporary Chinese word segmentation task [4]. The four tags of IBES represent the Beginning of a token, End of a token, Inside a token,[27] and Single unit,[28] respectively. As simplified versions, there are the IE and the IB tagging schemes, where only the end or the beginning of a token is addressed. In the work of Ding et al. [11], the IE scheme was used as a matter of fact. Basically, two different tags are sufficient for the tokenization task, while the performance of different tagging schemes may differ owing to the nature of specific languages. There can be more complex schemes for further classification of the I tag [45], although as stated in Kudo and Matsumoto [23], a very complex tagging scheme may lead to more "illegal" combinations of tags,[29] which does not always help the performance. In our specific data, the average token length ranges between one and two syllables, so we did not further classify the I tag using a more complex scheme. Only the IBES, IE, and IB schemes are compared in the experiments. Figure 6 shows an IBES tagging scheme for tokenization of the final part of the sentence in Figure 3. The IBES tags can be further attached with POS tags for joint tokenization and POS-tagging, as in the two lower rows in Figure 6. Table 5 lists the three feature templates we

[26] There are around 4,000 types in ALT data. A general estimation can be derived from syllable-based indexing of Burmese dictionaries. For example, in Okell and Allott [33], the entries are indexed by three-levels of 33 initial consonant letters, 4 medial consonant letters, and 65 rhymes. One initial consonant letter can take zero to three medial consonant letters to form an onset, and the types of onset are around 100 if obscure ones are included. Consequently, 7,000 will be a quite loose upper bound of possible onset-rhyme combinations. Even more irregular spellings such as stacked letters are included, the total types of syllables hardly reach a magnitude of 10^4.

[27] That is, the unit is neither the beginning nor the end of a token. Instead of I, M for Middle is also used.

[28] That is, the unit is simultaneously the beginning and the end of a token.

[29] In the IBES scheme, the sequences of IB, IS, BB, BS, EI, EE, SI, and SE are impossible. In the IE and the IB schemes, all sequences are possible, except the final tag in a sequence cannot be I in the IE scheme and the first tag cannot be I in the IB scheme.

Table 5. Feature Templates with Different Context Window Sizes

	uni-gram	bi-gram	tri-gram	4-gram
2	S_0^0, S_{-1}^{-1}, S_1^1	S_{-1}^0, S_0^1		
3	S_0^0, S_{-1}^{-1}, S_1^1, S_{-2}^{-2}, S_2^2	S_{-1}^0, S_0^1	S_{-2}^0, S_{-1}^1, S_0^2	
4	S_0^0, S_{-1}^{-1}, S_1^1, S_{-2}^{-2}, S_2^2, S_{-3}^{-3}, S_3^3	S_{-1}^0, S_0^1	S_{-2}^0, S_{-1}^1, S_0^2,	S_{-3}^0, S_{-2}^1, S_{-1}^2, S_0^3

used in the experiments, where S_m^n stands for the syllable sequence of the relative indices within $[m, n]$. In the upper row of Figure 6 (`REL. IDX.`), an example of relative indices is given for tagging the syllable with the gray background.

We used the `CRF++` toolkit[30] consistently in the experiments in this section. Another popular off-the-shelf tool is the `CRFsuit`,[31] which is much faster than `CRF++`. In our preliminary experiments, we found that the `CRFsuit` and `CRF++` have comparable performance when using identical feature templates. However, `CRF++` further supports bi-gram features on output tags, which can lead to slightly increased performance by further "tons of distinct features." We find that adding non-lexicalized bi-gram features of output tags leads to a good trade-off in terms of performance and time/memory consumption during model training. Therefore, all experimental results reported in this section include the features of the syllables listed in Table 5 and the non-lexicalized bi-gram features of output tags.[32]

4.2 Evaluation

We use F-score consistently to evaluate and compare experimental results. Specifically, the tokens segmented in a Burmese string are compared with the manual tokenization in test data. The F-score then is calculated as the harmonious average of the precision and recall in terms of tokens. The accuracy of tagging basic units, namely, syllables, is also presented as an auxiliary measure, which is not comparable across different output tagging schemes. As the development data are not required for training CRFs, the development data in Tables 2 and 3 are added to the training data in all CRF experiments. We also varied the training data size, to investigate how the quantity of training data affects the performance on test data. Specifically, the training data are halved gradually until around 1,000 sentences, that is, up to 1/16 of the ALT data and 1/8 of the CICLING data. In the presentation of this article, we use tables to report the numerical results using full training data, and use figures to illustrate the change in performance under different training data sizes.

Table 6 summarizes the main results of processing short tokens in the ALT test data using the full training data with different feature-tag combinations. The column of **separate tokenization** contains the results of plain tokenization, where no POS information is used. The column of **joint tokenization and POS-tagging** contains the results of simultaneously generated tokens and POS tags, where the tokenization tags are combined with POS tags to form a larger tagset. The F-score of bare tokens (without POS tags) and of joint tokens and POS tags (**token/POS**, as in Neubig et al. [31]) are listed, respectively, in this column. Compared with separate tokenization, the gains on the F-score of tokens obtained by joint tokenization and POS-tagging is also presented (Δ_{joint}). A noticeable phenomenon in Table 6 is that the `IE` and the `IB` schemes have much higher accuracy than the `IBES` scheme, while F-score is obviously lower. Clearly, the `IBES` scheme, though making

[30] http://taku910.github.io/crfpp/.
[31] http://www.chokkan.org/software/crfsuite/.
[32] The non-lexicalized bi-gram feature on output tags is annotated by a "B" according to the format of the `CRF++`'s template file. It is a feature of $\delta(y_{k-1}, y_k)$ in the notation used in this article.

the task more difficult, codes more useful information. Although impossible tagging sequences may appear in the IBES scheme, such inconsistencies appear rarely (once or twice) and only with smaller training data (1/16 and 1/8), which are negligible. As a minor fact, that IB is slightly better than IE in separate tokenization. A related phenomenon has been mentioned in Kudo and Matsumoto [23] that the IB scheme may reserve more information in segmenting continuous chunks, while our task is actually a chunking task on Burmese syllables.[33] Comparing the tokenization performance from the separate and joint processing, POS tag information can always boost tokenization performance, which is mentioned as a future work in Ding et al. [11] and is proved in this study. As the performance of tokenization in separate processing is already lower than that of joint processing, a two-pass first-tokenizing-then-POS-tagging processing cannot achieve better performance than the joint processing. As for the features, bi-grams on syllables (2−) seem adequate, and tri-grams (3−) bring limited gain, but additional features (4−) will lead to performance degradation owing to sparseness and over-fitting. The changes of F-scores with different training data sizes in joint processing are graphed in Figure 7, where the right-most values are presented in the **token/POS** column in Table 6. Generally, there is still considerable scope to improve performance by using more additional data, where the 4-gram features may boost performance more significantly.

Table 6 and Figure 7 present a basic group of the experimental results. Table 7 and Figure 8 contain the corresponding results of long tokens of the ALT data. Table 8 and Figure 9 contain the corresponding results of the CICLING data. The general phenomena and conclusions pertaining to the basic group can also be observed and concluded in two additional groups of experimental results. Specifically, the numerical results of long tokens in the ALT data are lower than those of short tokens. Notice the average length is 1.59 syllables for short tokens but 2.12 for long tokens. Therefore, bi-gram features are adequate for covering the range of short tokens but is relatively insufficient for long ones. High-order features, however, still do not increase performance due to sparseness. Consequently, the processing of long tokens becomes more difficult than that of short tokens. In the CICLING data, a difference from the ALT data is that the gain on tokenization performance due to joint processing is not as obvious as that in the ALT data. This may be ascribed to the POS tag scheme, which have been discussed in Section 3.2. Consequently, POS tags do not provide information consistently for tokenization and, thus, lead to limited improvement in the CICLING data.

There is two-layer information in the notation of the ALT data, while short and long tokens are processed separately in the illustrated experimental results. Table 9 provides further results related to two-pass processing between the two-layer annotations. In this group of experiment, the IBES scheme is applied, because it has been proved to be informative and efficient in previous results in this article. Table 9 presents the results of short-to-long and long-to-short generation, respectively, where SYL−TOK means the processing is based on syllables, that is, the syllables with labeled information of short (long) tokens are relabeled to generate long (short) tokens; TOK−TOK means the labeling is based directly on the generated short tokens where the short tokens are labeled to compose long tokens. Therefore, TOK−TOK is invalid for long-to-short generation. The accuracy in Table 9 is the labeling precision in the second pass, so the numerical results present rather high values. The F-score is the final performance of two-pass processing, where the automatically generated results with noise (error) in the first pass are reprocessed using the models trained in the second pass. The $\mathbf{\Delta_{2-pass}}$ shown in brackets are compared with corresponding one-pass results in Tables 7 and 6, respectively. Although not by a large margin, the two-pass processing from short tokens to long tokens is better than the one-pass processing directly from unlabeled syllables to

[33] So the IE scheme used in Ding et al. [11] is the worst option.

Table 6. Performance of CRFs in Processing Short Tokens in the ALT Data

feature-tag	separate tokenization		joint tokenization and POS-tagging			
	accuracy	F-score	accuracy	F-score		
		token		token	(Δ_{joint})	token/POS
2-IBES	94.9%	0.943	93.9%	**0.947**	(+0.004)	**0.940**
3-IBES	95.1%	0.944	93.9%	0.946	(+0.002)	**0.940**
4-IBES	94.9%	0.943	93.7%	0.945	(+0.002)	0.938
2-IE	96.3%	0.922	94.9%	0.938	(+0.016)	0.931
3-IE	96.9%	0.935	95.3%	0.943	(+0.008)	0.937
4-IE	96.9%	0.935	95.2%	0.943	(+0.008)	0.936
2-IB	96.7%	0.929	94.8%	0.939	(+0.010)	0.932
3-IB	97.0%	0.937	95.0%	0.942	(+0.005)	0.935
4-IB	97.0%	0.937	95.0%	0.942	(+0.005)	0.936

Table 7. Performance of CRFs in Processing Long Tokens in the ALT Data

feature-tag	separate tokenization		joint tokenization and POS-tagging			
	accuracy	F-score	accuracy	F-score		
		token		token	(Δ_{joint})	token/POS
2-IBES	95.5%	0.932	94.3%	**0.937**	(+0.005)	**0.929**
3-IBES	95.6%	0.933	94.3%	0.936	(+0.003)	**0.929**
4-IBES	95.4%	0.930	94.2%	0.934	(+0.004)	0.927
2-IE	96.4%	0.897	95.1%	0.921	(+0.024)	0.913
3-IE	97.3%	0.922	95.5%	0.931	(+0.009)	0.924
4-IE	97.2%	0.921	95.6%	0.930	(+0.009)	0.923
2-IB	97.0%	0.914	95.1%	0.927	(+0.013)	0.919
3-IB	97.4%	0.925	95.3%	0.930	(+0.005)	0.923
4-IB	97.4%	0.925	95.4%	0.931	(+0.006)	0.924

Table 8. Performance of CRFs in Processing the CICLING Data

feature-tag	separate tokenization		joint tokenization and POS-tagging			
	accuracy	F-score	accuracy	F-score		
		token		token	(Δ_{joint})	token/POS
2-IBES	95.5%	0.950	93.4%	**0.951**	(+0.001)	**0.934**
3-IBES	95.6%	0.951	93.4%	0.952	(+0.001)	0.933
4-IBES	95.2%	0.947	92.9%	0.947	(+0.000)	0.928
2-IE	96.8%	0.933	94.2%	0.942	(+0.009)	0.924
3-IE	97.3%	0.943	94.4%	0.946	(+0.003)	0.927
4-IE	97.1%	0.940	93.8%	0.940	(+0.000)	0.921
2-IB	97.0%	0.937	94.4%	0.944	(+0.007)	0.926
3-IB	97.3%	0.943	94.4%	0.947	(+0.004)	0.929
4-IB	97.2%	0.940	94.0%	0.943	(+0.003)	0.923

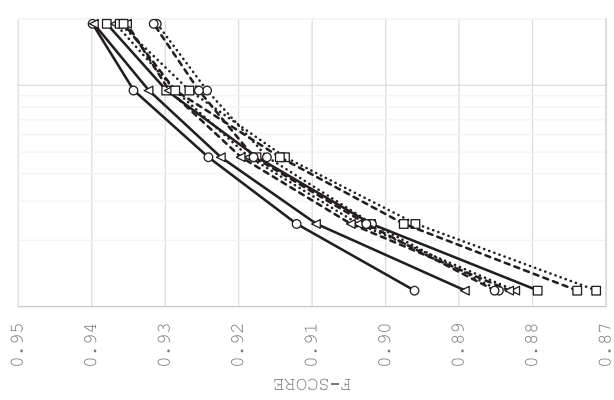

Fig. 7. Change of **F-score** of **token/POS** under **joint tokenization and POS-tagging** in Table 6 with different training data sizes.

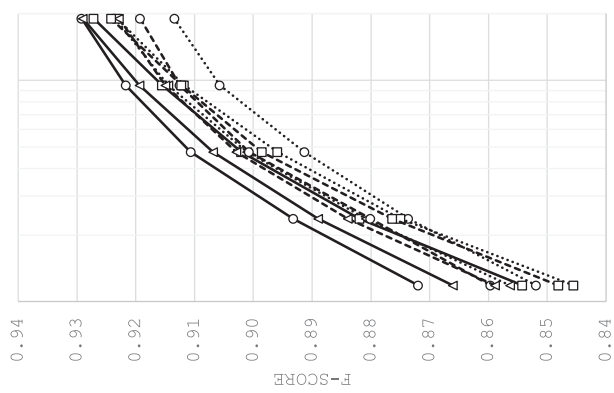

Fig. 8. Change of **F-score** of **token/POS** under **joint tokenization and POS-tagging** in Table 7 with different training data sizes.

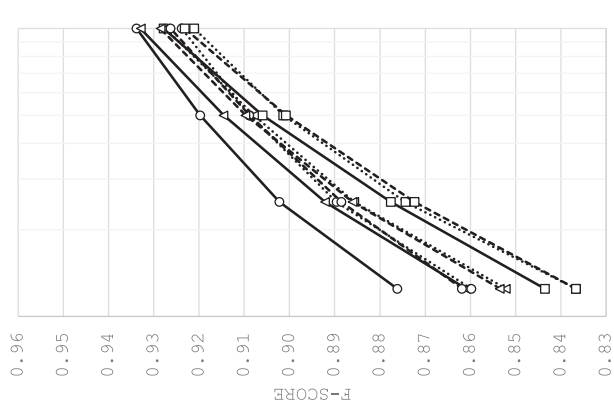

Fig. 9. Change of **F-score** of **token/POS** under **joint tokenization and POS-tagging** in Table 8 with different training data sizes.

Table 9. Performance of CRFs between Short and Long Tokens in the ALT Data

feature-tag.pass	short-to-long generation			long-to-short generation		
	accuracy	F-score		accuracy	F-score	
		token/POS	($\Delta_{2-\text{pass}}$)		token/POS	($\Delta_{2-\text{pass}}$)
2-IBES.SYL-TOK	99.1%	0.930	(+0.001)	98.5%	0.939	(−0.001)
3-IBES.SYL-TOK	99.1%	0.930	(+0.001)	98.5%	0.939	(−0.001)
4-IBES.SYL-TOK	99.1%	0.928	(+0.001)	98.5%	0.936	(−0.002)
2-IBES.TOK-TOK	98.9%	**0.931**	(+0.002)	–	–	–
3-IBES.TOK-TOK	98.9%	0.930	(+0.001)	–	–	–
4-IBES.TOK-TOK	98.8%	0.927	(+0.000)	–	–	–

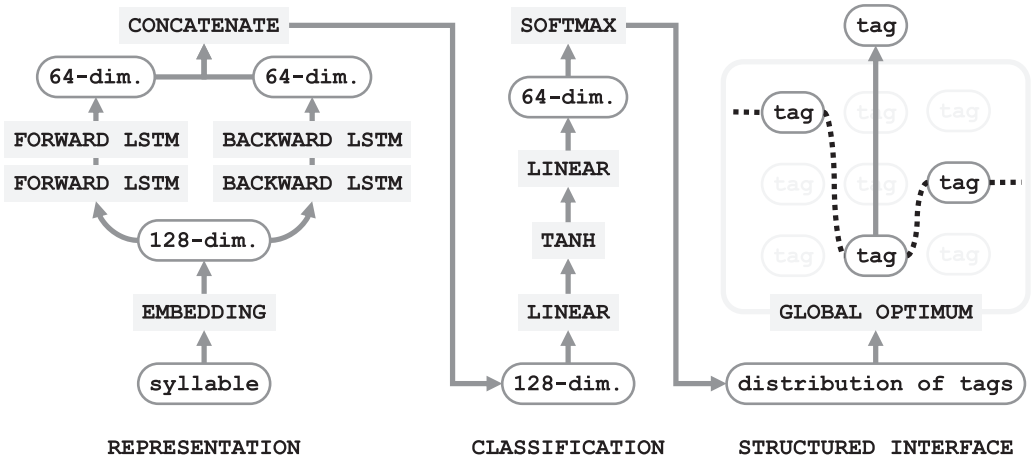

Fig. 10. Configuration of the LSTM-based RNN used in Burmese morphological analysis. (n-dim. stands for a vector in $\mathcal{R}^n$)

long tokens. Moreover, token-based two-pass processing is better than the syllable-based one. As for a reasonable explanation, the information of short tokens, even though not accurate, can help increase the performance of long tokens; short token-based features can cover a longer context, which leads to better results than those obtained using syllable-based features. Another benefit of short token-based processing for long tokens is that the boundary will always be consistent in two-layer processing, that is, the boundary of long tokens will always be the boundary of short tokens. However, long-to-short generation cannot provide better performance. We consider that it is natural to build long units from short units gradually, but not in a reversed manner.

5 BURMESE TOKENIZATION AND POS-TAGGING BY RNN

5.1 Network Structure, Ensemble, and Implementation

In the previous section, the experimental results obtained by CRF-based Burmese tokenization and POS-tagging were illustrated and investigated. In this section, we adopt the LSTM-based RNN approach. We first describe the network structure and the specific approaches applied before the experiment-based evaluation. The overall network structure is illustrated in Figure 10, where the three modules of representation, classification, and structured interface comprise a standard configuration for structured learning in many NLP tasks.

Specifically, the representation module is an automatic feature extractor, where discrete tokens (i.e., syllables) are encoded into an $\mathcal{R}^n$ space as an efficient representation. As the LSTM unit is used in a bidirectional manner, the embedding actually encoding global contextual features for each local token. In a relatively early attempt of Collobert et al. [6], this module is realized in a convolutional way over N-gram feature. In recent studies, encoding by bidirectional LSTM units has been a more standard way in various NLP tasks (e.g., Chen et al. [4]). The extracted features are then fed into a plain fully connected feed-forward NN, which performs as a non-linear classifier. The non-linear transformation in this module may be truncated [4] where all non-linear transformation was done by LSTM units. However, such a separate and simple classification module can make the configuration more trainable.[34] The final structured interface provide a global search on the sequences of output tags, which further models the dependent features of neighboring tags. As mentioned, the LSTM units have encoded global features into each token representation. If the LSTM layer can provide perfect contextual features, then the module of global search is also unnecessary. However, based on the same reason of the classification module, applying a configuration with clear functional division benefits the training process, where the LSTM layer, which is powerful and difficult to train, is restricted to feature extraction only.

More modifications can be added to the overall configuration, such as by further using CNN in the representation module, or by integrating a CRF-based module with the structured interface [26]. We applied the most general structure used in Figure 10, with details based on several observations from our preliminary experiments.

- The structured interface can accelerate convergence in training, but it cannot improve the performance substantially when using our data. We observed that even without the structured interface, the performance achieved on the development data is comparable to that achieved using the structured interface, requiring a greater number of training iterations. We attribute this to the ability of the representation module, using which contextual information can be well represented for the classification to compensate for the absence of structured learning. Thus, in the structured interface, we applied only a standard Viterbi algorithm to model the relationship between neighboring tags, which is adequate for the task.
- The number of LSTM layers in the representation module affects the performance, and only one layer is inadequate. We used two-layer bidirectional LSTM, which offers a trade-off between performance and training speed. A third layer cannot bring as much improvement as the second layer does, but it does increase the training time. Moreover, we tried other light-weight nonlinear units such as gated recurrent units [5] but there was little difference in performance. In practice, we applied a compact variant of LSTM with peepholes [15].
- The type of nonlinear function used in the classification module does not affect the performance considerably. Therefore, we applied the most common tanh function.
- The dimension of each layer affects the performance. The dimensions shown in Figure 10 are selected by arriving at a trade-off between training time and performance.
- As shown in Figure 10, the embedding is based on single syllables, that is, uni-gram of syllables. Higher orders such as bi- or tri-gram of syllables make the training slower without yielding obvious performance gains. Moreover, we tried pre-trained embedding, which did not lead to performance gains but it did accelerate convergence.

In practice, the training of the network is quite unstable. Even when we used development data to control over-fitting by selecting models having good performance on the development

[34]Notice in Chen et al. [4], stacked LSTMs cannot bring improvement, as they are too complex non-linear units to train.

data, the final performance on test data still diverged considerably with different initializations. This may be caused by the training data size, which is insufficient for NN-based processing. After many attempts, we discovered a practical ensemble method to combine a large amount of independently fast-trained networks to alleviate the instability in performance caused by initialization. The method is also efficient and robust from the viewpoint of improving the performance achieved by single networks. Specifically, we trained different networks separately with different initializations and only with a few iterations and then combined the results by a voting ensemble. The basic idea is to release the difficulties in tuning one refined model by using multiple roughly tuned models. It is also a practical solution in terms of computing resources. Fast training of multiple networks can be conducted in parallel if there are plenty of computing resources, or the networks can be trained one by one successively to increase the performance of the ensemble gradually when using limited computing resources. The details in the ensemble are as follows.

- For training single networks, we applied one initial iteration without the structured interface and then applied several further iterations with Viterbi searching, specifically, three further iterations for tokenization and five further iterations for joint tokenization and POS-tagging. We found that performance improvement on the development data was most obvious in these iterations. As we have not pursued refined models in the ensemble, this type of fast training is adequate for the ensemble.
- The ensemble was conducted by simple voting on each output tag, to select the most common one from the results of multiple models. As the selection is point-wise, it may generate illegal tag sequences.[35] This problem is not serious, and under a 100-model ensemble, less than 0.1% of the sequences generated in tokenization were illegal and less than 0.2% in joint tokenization and POS-tagging. We applied a straightforward solution to collect all possible bi-grams on output tags from the training data and selected only the possible sequences in the ensemble.[36]
- We tried a few more complex ensemble schemes, such as adding weights to different models, but these attempts did not yield better results. Because all models used in the ensemble are identical in mechanism, we believe that treating them in a simple manner as equals is the most mature solution.

The aforementioned model structure and ensemble processing were tested and selected based on their performance on the development data in Tables 2 and 3, and only the corresponding training data were used in model training for evaluation. We used the `DyNet` toolkit (version 2.0) [30] in the implementation. Specifically, model parameters are initialized by Xavier initialization [16] and learned by `Adam` [21] in the experiments, after trying several parameter-optimizing approaches that did not differ much in performance. We did not use dropout [37], because it decelerated the training and its effect was not as significant as that brought by the model ensemble. Hyper-parameters used in experiments are summarized in Table 10. On one GPU of Tesla K80, the training time for one tokenization model was around 20min and that for one joint tokenization and POS-tagging model was around one hour, with the described times of iterations using all of the ALT training data. We experimented with the ensemble of 100 models, which took approximately a day and half for

[35] Given the use of the structured interface in single-model training, illegal tag sequences hardly occurred from the results by single models except trained on a very few training data.

[36] Therefore, the possible sequences may change along with the training data size. In tokenization, there is no effect, because there are only four `IBES` tags. In joint tokenization and POS-tagging, smaller training data may contain fewer variants in combination. However, as mentioned, this is not a serious problem, and it affects the numerical results negligibly.

Table 10. Hyper-parameters for LSTM-based RNN

hyper parameter	value	note
Dimension of embedding	128	shown in Figure 10
Dimension of forward / backward LSTM states	64/64	
Dimension of feed-forward classification output	64	
`Adam`'s learning rate α	10^{-3}	default by `DyNet`
`Adam`'s moving average β_1 for the mean	0.9	
`Adam`'s moving average β_2 for the variance	0.999	
`Adam`'s bias ϵ	10^{-8}	
Dropout rate	–	disabled
Ensemble size	5, 10, 20, 50, and 100	

tokenization model training and four days for joint tokenization and POS-tagging model training, when training 100 models serially.

5.2 Evaluation

The metrics used here are identical to those in the CRF evaluation. The training data are also halved gradually for comparison. Given that a small portion of the development data has been used in model selection rather than in model training, the training data are slightly smaller than those used in the CRF experiments. The tagging scheme of `IBES` is used consistently in all the experiments of LSTM-based RNN, because it is the most efficient scheme, as shown in previous experiments. The number of models in the ensemble are also compared to investigate the effect of ensemble size.

Table 11 is the LSTM-based RNN version of Table 6, with the accuracy on both test and development (**dev.**) data. The results of the ensemble (`ENS-`) over 5, 10, 20, 50, and 100 models are illustrated. The best (`MAX@100`) and the worst (`MIN@100`) single models among the total of 100 models are listed as well. The results of `2-` and `3-IBES` in Table 6 are listed in Table 11 for comparison. The comparisons of `ENS-100` of LSTM-based RNN in Table 11 with the CRF results in Table 6 are graphed in Figure 11.[37] Generally, LSTM-based RNN boosted by ensemble can achieve performance comparable to that of CRFs on the full training data, where the ensemble can lead to a certain gain in performance. In joint tokenization and POS-tagging, the LSTM-based RNN has better performance on small training data compared to the case of separate tokenization. Therefore, it is obvious that a more informative output tagset has more significant effects, especially on small training data, because the sparseness of the discrete features used in the CRFs is alleviated by the embedding to a dense representation in a low-dimension real space facilitated by RNN. As for the effect of the ensemble, it can improve the performance of only a few models (e.g., five), although there is gradual and steady but insignificant improvement in the performance of a large number of models.

As in the CRF evaluation, Table 11 presents a basic group of the experimental results obtained using LSTM-based RNN. Table 12 is the LSTM-based RNN version of Table 7, which presents the experimental results of long tokens in the ALT data. Table 13 is the LSTM-based RNN version of Table 8, which presents the experimental results obtained using the CICLING data. The the effect of training data size on Tables 12 and 13 are graphed in Figures 12 and 13, respectively. From the further two groups of experiments, we can observe that LSTM-based RNN also performs better

[37] Because `4-IBES` never achieves better results, it is omitted from tables, and illustrated only in figures.

Table 11. Performance of LSTM-based RNN in Processing Short Tokens in the ALT Data

model	separate tokenization			joint tokenization and POS-tagging				
	accuracy		F-score	accuracy		F-score		
	test	(dev.)	token	test	(dev.)	token	(Δ_{joint})	token/POS
RNN ENS-5	94.8%	(95.1%)	0.942	93.7%	(93.8%)	0.945	(+0.003)	0.938
RNN ENS-10	94.9%	(95.1%)	0.943	93.7%	(93.9%)	0.945	(+0.002)	0.938
RNN ENS-20	95.0%	(95.1%)	0.943	93.8%	(94.0%)	0.946	(+0.003)	0.939
RNN ENS-50	95.0%	(95.1%)	0.944	93.9%	(94.1%)	0.946	(+0.002)	0.939
RNN ENS-100	95.1%	(95.1%)	0.944	93.9%	(94.1%)	**0.947**	(+0.003)	**0.940**
RNN MAX@100	94.4%	(94.6%)	0.938	93.1%	(93.3%)	0.939	(+0.001)	0.931
RNN MIN@100	93.9%	(94.2%)	0.932	92.6%	(92.9%)	0.937	(+0.005)	0.928
CRF 2-IBES	94.9%	–	0.943	93.9%	–	**0.947**	(+0.004)	**0.940**
CRF 3-IBES	95.1%	–	0.944	93.9%	–	0.946	(+0.002)	**0.940**

Table 12. Performance of LSTM-based RNN in Processing Long Tokens in the ALT Data

model	separate tokenization			joint tokenization and POS-tagging				
	accuracy		F-score	accuracy		F-score		
	test	(dev.)	token	test	(dev.)	token	(Δ_{joint})	token/POS
RNN ENS-5	95.6%	(95.7%)	0.933	94.1%	(94.3%)	0.934	(+0.001)	0.926
RNN ENS-10	95.6%	(95.9%)	0.933	94.2%	(94.5%)	0.935	(+0.002)	0.928
RNN ENS-20	95.7%	(95.9%)	0.934	94.3%	(94.7%)	0.935	(+0.001)	0.928
RNN ENS-50	95.7%	(96.0%)	0.935	94.4%	(94.7%)	**0.937**	(+0.002)	**0.930**
RNN ENS-100	95.7%	(96.0%)	0.935	94.4%	(94.6%)	**0.937**	(+0.002)	**0.930**
RNN MAX@100	95.1%	(95.4%)	0.926	93.7%	(93.8%)	0.928	(+0.002)	0.920
RNN MIN@100	94.6%	(95.0%)	0.919	93.0%	(93.4%)	0.922	(+0.003)	0.913
CRF 2-IBES	95.5%	–	0.932	94.3%	–	**0.937**	(+0.005)	0.929
CRF 3-IBES	95.6%	–	0.933	94.3%	–	0.936	(+0.003)	0.929

Table 13. Performance of LSTM-based RNN in Processing the CICLING Data

model	separate tokenization			joint tokenization and POS-tagging				
	accuracy		F-score	accuracy		F-score		
	test	(dev.)	token	test	(dev.)	token	(Δ_{joint})	token/POS
RNN ENS-5	94.8%	(95.4%)	0.942	93.1%	(93.3%)	0.950	(+0.008)	0.930
RNN ENS-10	95.0%	(95.4%)	0.944	93.4%	(93.6%)	0.951	(+0.007)	0.933
RNN ENS-20	95.1%	(95.5%)	0.946	93.4%	(93.9%)	0.950	(+0.004)	0.933
RNN ENS-50	95.2%	(95.6%)	0.947	93.5%	(93.9%)	0.952	(+0.005)	0.934
RNN ENS-100	95.2%	(95.7%)	0.947	93.6%	(94.0%)	**0.953**	(+0.006)	**0.935**
RNN MAX@100	94.3%	(94.5%)	0.937	91.9%	(92.1%)	0.941	(+0.004)	0.918
RNN MIN@100	93.4%	(93.8%)	0.928	91.0%	(91.1%)	0.934	(+0.006)	0.909
CRF 2-IBES	95.5%	–	0.950	93.4%	–	0.951	(+0.001)	0.934
CRF 3-IBES	95.6%	–	0.951	93.4%	–	0.952	(+0.001)	0.933

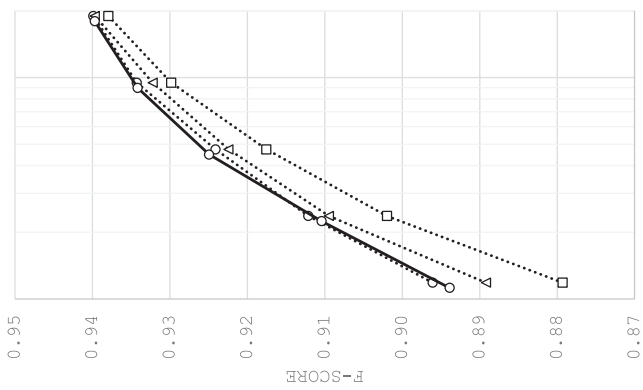

Fig. 11. Change of **F-score** of **token/POS** under **joint tokenization and POS-tagging** in Table 11 with different training data sizes. (comparison of RNN (ENS-100) and CRFs)

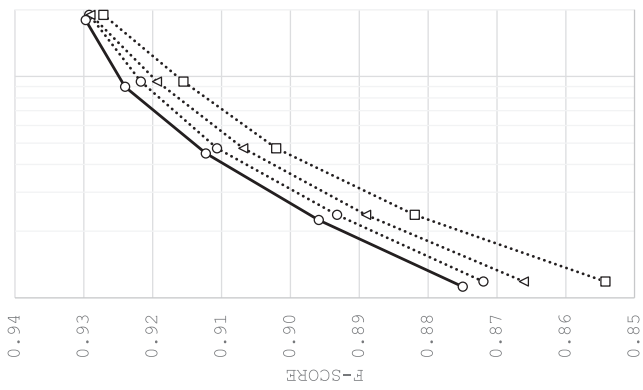

Fig. 12. Change of **F-score** of **token/POS** under **joint tokenization and POS-tagging** in Table 12 with different training data sizes. (comparison of RNN (ENS-100) and CRFs)

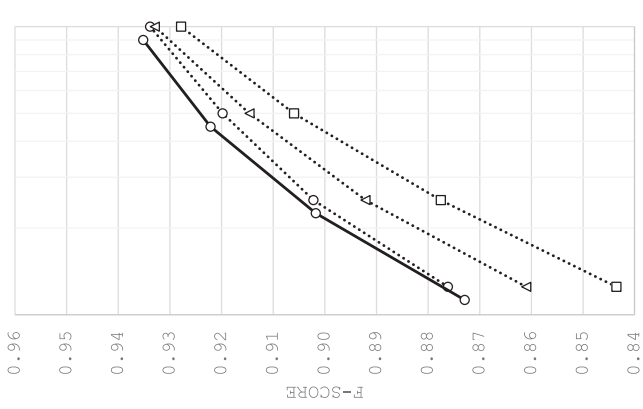

Fig. 13. Change of **F-score** of **token/POS** under **joint tokenization and POS-tagging** in Table 13 with different training data sizes. (comparison of RNN (ENS-100) and CRFs)

Table 14. Performance of of LSTM-based RNN in Short-to-Long Token Processing in the ALT Data

model	syllable-to-token				token-to-token			
	accuracy		F-score		accuracy		F-score	
	test	(dev.)	token/POS	($\Delta_{2-\text{pass}}$)	test	(dev.)	token/POS	($\Delta_{2-\text{pass}}$)
RNN ENS-5	99.0%	(99.0%)	0.928	(+0.002)	98.9%	(98.9%)	0.929	(+0.003)
RNN ENS-10	99.0%	(99.1%)	0.928	(+0.000)	98.9%	(98.9%)	0.928	(+0.000)
RNN ENS-20	99.1%	(99.1%)	0.928	(+0.000)	98.9%	(98.9%)	0.928	(+0.000)
RNN ENS-50	99.1%	(99.1%)	0.928	(−0.002)	98.9%	(99.0%)	0.928	(−0.002)
RNN ENS-100	99.1%	(99.1%)	0.928	(−0.002)	98.9%	(99.0%)	0.928	(−0.002)
RNN MAX@100	98.9%	(99.0%)	0.927	(+0.007)	98.9%	(98.9%)	0.928	(+0.008)
RNN MIN@100	98.6%	(98.7%)	0.925	(+0.012)	98.6%	(98.6%)	0.926	(+0.013)
CRF 2-IBES	99.1%	–	0.930	(+0.001)	98.9%	–	**0.931**	(+0.002)
CRF 3-IBES	99.1%	–	0.930	(+0.001)	98.9%	–	0.930	(+0.001)

in joint tokenization and POS-tagging than in separate tokenization. The effect of RNN is more obvious than that of CRFs when processing long tokens in the ALT data. As discussed, low-order N-gram features cannot capture sufficient local information, while high-order N-gram features cause sparseness. This dilemma can thus be alleviated by the strength of RNN, which provides more efficient feature representation. The LSTM-based RNN also outperforms CRFs on joint tokenization and POS-tagging the CICLING data, which indicates that RNN can take advantage of relatively complex and inconsistent features more efficiently than CRFs.

The final group of experiments involves two-pass processing using LSTM-based RNN to generate long tokens from the results of short tokens in the ALT data. The generation of short tokens from the results of long tokens is neither natural nor efficient, as evidenced by the experiments of CRFs. Therefore, we do not continue to experiment in this way. The results of syllable-based processing and token-based processing are given in Table 14. In the token-based processing, we tried to use larger dimensions for different layers in the LSTM-based RNN than those in the syllable-based processing, but this did not yield a better result. The results in Table 14 are based on the exact network structure shown in Figure 10. In training the models for the ensemble, one iteration without Viterbi search plus three iterations with Viterbi search were applied for syllable-based processing and five iterations with Viterbi search for token-based processing. The performance is slightly lower than that of CRFs. The ensemble effect is not obvious in the two-pass processing experiments. Because the methodology of the NN-based approach is rooted in the ultimate joint model for end-to-end processing, the efficiency of RNN cannot be increased in such two-pass processing.

6 DISCUSSION

6.1 Error Analysis

In Sections 4 and 5, we have presented detailed numerical results to illustrate the performance and the characteristics of different approaches for Burmese tokenization and POS-tagging tasks. From a linguistic viewpoint, we analyze errors across different approaches based on their final results.

Tables 15, 16, and 17 list the frequency of different tags (#) in ALT and CICLING test data, with the error frequency and rate for each tag from result of joint tokenization and POS-tagging by different approaches. We compare the best results generated by CRFs with those of different LSTM-based RNN models. On the ALT data, most improvement on performance can be attributed to the improvement on identifying n-tagged tokens in multiple RNN-model ensemble. However,

Table 15. Error Analysis on Short Tokens in the ALT Data (from the Corresponding Results in Table 11)

		3-IBES		ENS-100		ENS-5		MAX@100		MIN@100	
tag	**#**	**#error**	**rate**	**#error**	**rate**	**#error**	**rate**	**#error**	**rate**	**#error**	**rate**
o-	14,609	198	1.4%	133	0.9%	145	1.0%	172	1.2%	188	1.3%
n	10,050	1,130	11.2%	1,165	11.6%	1,215	12.1%	1,342	13.4%	1,387	13.8%
v	5,764	609	10.6%	554	9.6%	572	9.9%	617	10.7%	633	11.0%
.	3,038	24	0.8%	25	0.8%	27	0.9%	25	0.8%	24	0.8%
1	1,143	46	4.0%	34	3.0%	38	3.3%	45	3.9%	33	2.9%
n-	821	28	3.4%	21	2.6%	28	3.4%	36	4.4%	27	3.3%
o	662	100	15.1%	98	14.8%	93	14.0%	100	15.1%	121	18.3%
a	440	93	21.1%	93	21.1%	93	21.1%	105	23.9%	107	24.3%
a-	275	3	1.1%	5	1.8%	6	2.2%	9	3.3%	9	3.3%
n-/o-	28	1	3.6%	2	7.1%	0	0.0%	0	0.0%	0	0.0%
total	36,830	2,232	–	2,130	–	2,217	–	2,451	–	2,529	–

Table 16. Error Analysis on Long Tokens in the ALT Data (2-IBES.TOK-TOK from Tables 9, Others from the Corresponding Results in Table 12)

		2-IBES.TOK-TOK		ENS-100		ENS-5		MAX@100		MIN@100	
tag	**#**	**#error**	**rate**	**#error**	**rate**	**#error**	**rate**	**#error**	**rate**	**#error**	**rate**
n	10,365	1,172	11.3%	1,229	11.9%	1,321	12.7%	1,366	13.2%	1,536	14.8%
o-	7,248	113	1.6%	113	1.6%	113	1.6%	143	2.0%	156	2.2%
v	3,116	308	9.9%	321	10.3%	325	10.4%	359	11.5%	396	12.7%
.	3,032	23	0.8%	21	0.7%	26	0.9%	31	1.0%	30	1.0%
a	1,911	176	9.2%	175	9.2%	178	9.3%	191	10.0%	220	11.5%
n-	770	29	3.8%	23	3.0%	25	3.2%	32	4.2%	34	4.4%
o	535	76	14.2%	80	15.0%	82	15.3%	105	19.6%	99	18.5%
1	460	39	8.5%	45	9.8%	51	11.1%	45	9.8%	47	10.2%
a-	275	5	1.8%	5	1.8%	5	1.8%	10	3.6%	8	2.9%
n-/o-	28	0	0.0%	2	7.1%	0	0.0%	0	0.0%	0	0.0%
total	27,740	1,941	–	2,014	–	2,126	–	2,282	–	2,526	–

RNNs cannot really beat CRFs on the performance of n-tagged tokens, but they usually have a better performance on v-tagged short tokens. On the CICLING data, the improvement from multiple RNN-model ensemble contributes more equally on different tags. An interesting phenomenon on the CICLING data is that the precision on fw tag, i.e., foreign word, is very low by CRFs but improved much by RNNs. This suggests that this tag is less related to local morphological features but more related to semantic features that can be captured better by the capacity of RNNs.

Generally, CRF models tend to produce less segmentation, i.e., longer tokens, while RNN models tend to offer a more fragmentary segmentation. We consider this is the underlying reason of the aforementioned observation. Within the experimental results, the numbers of tokens generated by CRFs are usually less than those in manual reference. Oppositely, RNNs always generate more tokens than those in manual reference. In the ALT data, the verbal constituents were manually segmented more fragmentary than nominal ones. Therefore, the RNNs outperforms CRFs on v-tagged short tokens in the ALT data. On the CICLING data, however, RNNs have better performance on nouns (n) while worse on verbs (v) compared with CRFs. We consider the reasons are

Table 17. Error Analysis on Tokens in the CICLING Data. (from the corresponding results in Table 13)

		2-IBES		ENS-100		ENS-5		MAX@100		MIN@100	
tag	**#**	**#error**	**rate**	**#error**	**rate**	**#error**	**rate**	**#error**	**rate**	**#error**	**rate**
n	5,652	598	10.6%	583	10.3%	603	10.7%	751	13.3%	811	14.3%
part	4,670	150	3.2%	138	3.0%	155	3.3%	179	3.8%	206	4.4%
ppm	3,400	51	1.5%	42	1.2%	47	1.4%	55	1.6%	62	1.8%
v	3,085	285	9.2%	296	9.6%	314	10.2%	351	11.4%	413	13.4%
punc	1,522	3	0.2%	0	0.0%	0	0.0%	0	0.0%	0	0.0%
conj	1,037	50	4.8%	39	3.8%	47	4.5%	84	8.1%	54	5.2%
adj	656	142	21.6%	127	19.4%	131	20.0%	151	23.0%	179	27.3%
num	365	10	2.7%	6	1.6%	4	1.1%	11	3.0%	12	3.3%
adv	264	44	16.7%	47	17.8%	51	19.3%	67	25.4%	74	28.0%
pron	247	28	11.3%	18	7.3%	21	8.5%	24	9.7%	24	9.7%
fw	187	113	60.4%	50	26.7%	61	32.6%	38	20.3%	60	32.1%
tn	180	14	7.8%	10	5.6%	11	6.1%	16	8.9%	15	8.3%
abb	20	0	0.0%	1	5.0%	1	5.0%	1	5.0%	1	5.0%
sb	18	6	33.3%	4	22.2%	4	22.2%	5	27.8%	5	27.8%
int	12	1	8.3%	3	25.0%	3	25.0%	3	25.0%	3	25.0%
total	21,315	1,495	–	1,364	–	1,453	–	1,736	–	1,919	–

(1) the nominal constituents in the CICLING data is less complex than those in the ALT data, and (2) the verbal constituents in the CICLING data are not annotated as fragmentary as those in the ALT data, and a portion of grammaticalized verbs are actually annotated as particles (part).

To further illustrate the difference in the behaviors of CRFs and RNNs, we show three examples in Figure 14. Because Burmese has a certain similarity to Chinese in morphology, comparison of Burmese and Chinese is offered in case for a better understanding. In the example of (I), the glosses for the four tokens in manual annotation are "*sport*" (1 အားကစား), "*many*" (2 အတော်များများ), "*to be absent*" (3 ပျက်စီးသွား), and a sentence ending suffix (4 သည်).[38] The result by CRF was identical to the manual annotation, while RNN provided a more fragmentary segmentation, where 2 အတော်များများ was further segmented into 2 အတော်များ 3 များ, and 3 ပျက်စီးသွား into 4 ပျက်စီး 5 သွား. The sequence of 2 အတော်များ 3 များ generated by RNN is improper although the action is understandable, as များ is a common nominal suffix to denote plurality. However, it is used in a more analytic way in this expression, where အတော် can be interpreted as "*very*" and များ here as "*many*." A reduplication of များ stresses the extent. Although such substructure of အတော်များများ can be identified, an ultimate segmentation as အတော် - များ - များ seems excessive. Comparing to Chinese, it also has a paradigm to form expressions in a pattern of A-B-B, where A and B are single characters. Such expression is commonly consider as one integrated word. Although native speakers can figure out the separate meaning of A and B, the meaning of B may be vague, which is exactly the case of this Burmese example. Another fragmentary segmentation of 4 ပျက်စီး 5 သွား generated by RNN provides a more fundamental phenomenon. The two tokens are actually independent verbal morphemes that ပျက်စီး of "*to fail*" and သွား of "*to go*," and they are combined here to form an emphasized expression. This combination can be compared

[38]The original English text is "… *a number of sports have been affected* …."

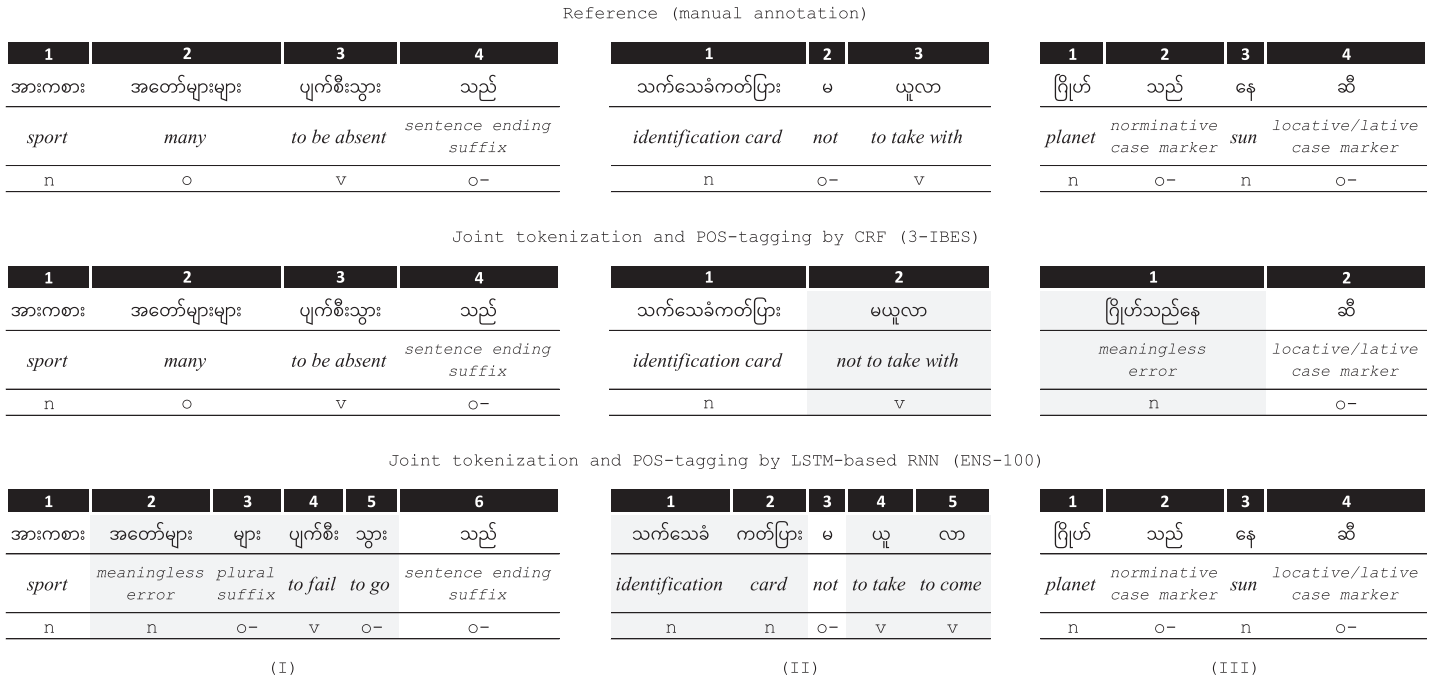

Fig. 14. Examples of short token annotation generated by CRF and LSTM-based RNN models, compared with the manual annotation. In each table, the index of words, the Burmese words, the corresponding International Phonetic Alphabet, the English glosses, and the nova tags are in rows from top to bottom.

with Chinese 失去 ("*to lose*") where 失 is "*to lose, to fail*" and 去 is "*to go, to leave.*" As a highly analytic language, such combination over common morphemes is frequent in Burmese and the stableness of the combination largely depends on contexts. The CRF's result is more intuitive to native speakers in this instance. A complex nominal constituent is presented in example (II). The glosses for the three tokens in manual annotation are "*identification card*" (❶ သက်သေခံကတ်ပြား), "*not*" (❷ မ), and "to take with" (❸ ယူလာ).[39] The သက်သေခံကတ်ပြား can be ultimately analyzed as four morphemes သက်သေ - ခံ - ကတ် - ပြား, which can be interpreted as "*witness-to receive-card-flat.*" The ကတ် is a transliteration of "*card*" and ပြား is a morpheme with vague meaning to describe flat objects. The expression of ကတ်ပြား can be used independently, which have a similar meaning and composition as Chinese 卡片, where 卡 is also from "*card*" and 片 for "*slice.*" The CRF generated the exact long token while RNN split it into two tokens of ❶ သက်သေခံ ❷ ကတ်ပြား., which is not wrong but more analytic. Considering the example is on short tokens, the RNN's output is actually more proper than the manual annotation, according to the annotation principles. In the second half of this example, CRF concatenated the ❷ မ ❸ ယူလာ in manual annotation into ❷ မယူလာ. This is an obvious mistake that the negation particle မ was not recognized. The same as the case in (I), RNN again generated a fragmentary segmentation as ❸ မ ❹ ယူ ❺ လာ. Here, ယူ has a meaning of "*to take*" and လာ "*to come.*" This instance can be compared with Chinese 带来 ("*to bring*") where 带 is "*to take*" and 来 is "*to come.*" The combination of 带 and 来 in Chinese is not so tight, that the expressions of "带 - *something* - 来" and "带来 - *something*" are both possible, with differences in nuance. However, Burmese is a head-final language that the word order should always be "*something* - ယူ - လာ". The strength of connection between ယူ and လာ are more difficult to judge only from superficial features, even for native speakers.[40] The example (III) is about unknown word. The glosses for the four tokens in manual annotation are "planet" (❶ ဂြိုဟ်), a nominative marker (❷ သည်), "sun" (❸ နေ), and a locative/lative case marker (❹ ဆီ).[41] The ဂြိုဟ် is a loan word from Sanskrit and is out of the vocabulary of training data. Generally, statistical models can identify unknown words to a certain extend by their contexts, but CRF failed in this instance. A possible reason is the trigger of the nominative case-marker သည် is not strong enough, noticing the identical token is also a common verbal suffix as in example (I). RNN offered the correct analysis here, attributed to its preference on fragmentary segmentation. The nominal and verbal suffixes are annotated identically by o– in the ALT data, because their roles can be identified by their preceding tokens (n or v). However, the preceding word here is out of the vocabulary of training data. If the nominal and verbal suffixes သည် are annotated separately, then distributions of the two types of သည် can be modeled more distinguishable, by which this failure by CRF will probably be avoided. The example (III) suggests that a more informative annotation may be beneficial to identify unknown words.[42]

From the statistics over entire test sets and the case study on typical examples, the behaviors of CRFs and RNNs are clearly illustrated. The discussion in this subsection mainly focuses on the linguistic features of Burmese. We provide discussions in a context of engineering practice in the following subsection.

[39] The original English text is "… *none … were carrying identification* …."
[40] More details on Burmese verbal expressions as shown in examples (I) and (II) can be referred to Bernot [3].
[41] The original English text is "… *the planet … to the sun* …."
[42] As most unknown words are nouns, another reasonable solution is the mentioned large dictionary.

6.2 Data and Approach

Temporarily, the size of training data is still the most crucial factor affecting performance in Burmese tokenization and POS-tagging. The ALT data contain around 0.66 $\mathcal{M}$ short tokens and 0.50 $\mathcal{M}$ long tokens; the CICLING data contain only around 0.19 $\mathcal{M}$ tokens. By referring to several popular Chinese word segmentation datasets presented in Emerson [12], we can find the scale on words (tokens) ranges from 1.10 $\mathcal{M}$ (*Peking University corpus*) to 5.45 $\mathcal{M}$ (*Academia Sinica corpus*), that is, there is one order of magnitude difference in the quantity, despite the fact that the ALT Burmese data contain more POS-tagging information than only tokenization. A more comparable corpus is *The Balanced Corpus of Contemporary Written Japanese*,[43] which is characterized by two-layer well-designed annotation [32] and contains over one hundred million words. However, such a huge corpus is based on more than thirty years of NLP development in the Japanese community. In terms of the research community for Burmese processing, the ALT Burmese data prepared in this study provide a solid basis for further development in coming years.

Although RNNs achieved better performance than CRFs by a small margin in some cases in experimental results, we believe that the traditional CRFs continue to constitute a more realistic approach than RNN under the current circumstance, considering (1) the quantity of the annotated resources, (2) interpretability of the model, and (3) the requirement of computing resources. The empirically based methodology of NN-based approaches can offer efficient end-to-end solutions, where the human-designed features are largely (if not completely) substituted by huge amounts of data and strong computing resources. However, Burmese processing is still in the early stage where more practice-based investigations are required. As a mature technique, CRFs can be implemented, applied, and maintained more feasibly than RNNs. CRFs can also be extended to semi-supervised learning interfaces [14] to make use of large amounts unlabeled data, which accommodates the temporary case of Burmese.

As discussed in Section 4.1, the processing of Burmese is totally syllable-based in this study. Further phonology-related sub-syllabic structures can be deducted in Ding et al. [10]. We tried a few sub-syllabic features in CRFs and added character embedding to RNNs, but these attempts did not yield better performance. Specific sub-syllabic features may offer certain information about etymology, but they hardly contribute to the precision of automatic processing.[44] As mentioned, Burmese syllables can be compared with Chinese characters, which are relatively independent writing units, we consider the syllable should be the natural and standard unit in Burmese processing, independent from specific machine learning approaches.

7 CONCLUSION AND FUTURE WORK

In this study, we focused on two primary tasks in Burmese morphological analysis: tokenization and POS-tagging, from annotated corpus preparation to experiment-based investigation. Our annotated corpus of 20,000 Burmese sentences has been released under a `CC BY-NC-SA` license to the research community. We conducted experiments by using the standard sequence-labeling approach of CRFs and a state-of-the-art LSTM-based RNN approach. The investigations and discussions in this study provide a solid basis for further research and development in Burmese processing. We conclude the contribution of this study as follows.

[43] http://pj.ninjal.ac.jp/corpus_center/bccwj/en/.

[44] Taking English as an example, a word beginning with *wh-* is likely to have a Germanic origin, while *rh-* can strongly indicate the word is from a Greek root. However, such clues do not directly contribute to identify morphological or syntactic roles of specific words.

- The released corpus is the largest open-access database of annotated Burmese when this manuscript was prepared in 2017. This corpus will accelerate research and development of Burmese textual processing techniques.
- This study presented comprehensive experimental results and analyses on Burmese tokenization and POS-tagging. The efficiency of joint processing of the two tasks have been shown. Engineering issues such as feature selection and model ensemble are also addressed.
- This work on Burmese presents an example on establishing basic annotated data and investigating proper NLP techniques from the very beginning for a low-resourced language. The experiences obtained in this work will be potentially helpful for other low-resourced language processing.

We have two clear directions for future work. Regarding Burmese processing, we have mentioned the future plan about annotated data development in Section 3.3, including the application of universal POS tags and the development of large-scale neologism dictionary. Ultimately, a Burmese treebank will be built on the morphologically annotated sentences prepared in this study. The trio of tokenization, POS-tagging, and syntactic parsing will become a complete cornerstone for Burmese processing. As for the scope of NLP on low-resourced languages in Asia, the next language on our schedule is Khmer. The experiences gained in developing data and techniques for Burmese processing can help us annotate and process the Khmer language efficiently.

ACKNOWLEDGMENTS

We thank all the translators and annotators contributing in the construction of Burmese ALT data. We thank Dr. Zaw Myint, Director General of the *Department of Myanmar Nationalities' Languages, Ministry of Education, Myanmar*, and Dr. Mie Mie Thet Thwin, rector of the *University of Computer Studies, Yangon*, for their help in verifying the annotated Burmese ALT data. We thank Dr. Atsushi Fujita, and those professional anonymous reviewers for their valuable comments and suggestions. This work was conducted under the program *"Promotion of Global Communications Plan: Research, Development, and Social Demonstration of Multilingual Speech Translation Technology"* of the *Ministry of Internal Affairs and Communications, Japan.*

REFERENCES

[1] Aye Myat Mon, Soe Lai Phyue, Myint Myint Thein, Su Su Htay, and Thinn Thinn Win. 2010. Analysis of Myanmar word boundary and segmentation by using statistical approach. In *Proceedings of the ICACTE*. 233–237.

[2] Vincent Berment. 2004. *Methods to Computerize "Little Equipped" Languages and Groups of Languages*. Ph.D. Dissertation.

[3] Denise Bernot. 1980. *Le prédicat en birman parlé*, vol. 8. Peeters Publishers.

[4] Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015. Long short-term memory neural networks for Chinese word segmentation. In *Proceedings of the EMNLP*. 1197–1206.

[5] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the EMNLP*. 1724–1734.

[6] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* 12 (2011), 2493–2537.

[7] Department of the Myanmar Language Commission. 2014. *Myanmar-English Dictionary (Myanma-anggalip Abidan)* (12th ed.). Ministry of Education, the Republic of the Union of Myanmar.

[8] Department of the Myanmar Language Commission. 2016. *Myanmar Grammar (Myanma Sadda)* (3rd ed.). Ministry of Education, the Republic of the Union of Myanmar (in Burmese).

[9] Chenchen Ding, Masao Utiyama, and Eiichiro Sumita. 2018. NOVA: A feasible and flexible annotation system for joint tokenization and part-of-speech tagging. *ACM Trans. Asian Low-Resour. Lang. Info. Process.* 18, 2 (2018), 17.

[10] Chenchen Ding, Win Pa Pa, Masao Utiyama, and Eiichiro Sumita. 2017. Burmese (Myanmar) name romanization: A sub-syllabic segmentation scheme for statistical solutions. In *Proceedings of the PACLING*. 227–238.

[11] Chenchen Ding, Ye Kyaw Thu, Masao Utiyama, and Eiichiro Sumita. 2016. Word segmentation for Burmese (Myanmar). *ACM Trans. Asian Low-Resour. Lang. Info. Process.* 15, 4 (2016).
[12] Thomas Emerson. 2005. The second international Chinese word segmentation bakeoff. In *Proceedings of the SIGHAN*. 123–133.
[13] Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Proceedings of the EACL*. 173–179.
[14] Ryo Fujii, Ryo Domoto, and Daichi Mochihashi. 2017. Nonparametric Bayesian semi-supervised word segmentation. *Trans. Assoc. Comput. Linguist.* 5 (2017), 179–189.
[15] Felix A. Gers, Nicol N. Schraudolph, and Jürgen Schmidhuber. 2002. Learning precise timing with LSTM recurrent networks. *J. Mach. Learn. Res.* 3 (Aug.2002), 115–143.
[16] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the AISTATS (PMLR)*, vol. 9. 249–256.
[17] Klaus Greff, Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. 2017. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* (2017), 1735–1780.
[18] Hla Hla Htay, G. Bharadwaja Kumar, and Kavi Narayana Murthy. 2007. *Statistical Analyses of Myanmar Corpora*. Technical Report. Department of Computer and Information Sciences, University of Hyderabad.
[19] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9, 8 (1997), 1735–1780.
[20] Khin War War Htike, Ye Kyaw Thu, Zuping Zhang, Win Pa Pa, Yoshinori Sagisaka, and Naoto Iwahashi. 2017. Comparison of six POS tagging methods on 10K sentences Myanmar language (Burmese) POS tagged corpus. In *Proceedings of the CICLING*.
[21] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the ICLR*.
[22] Taku Kudo and Yuji Matsumoto. 2001. Chunking with support vector machines. In *Proceedings of the NAACL*. 1–8.
[23] Taku Kudo and Yuji Matsumoto. 2002. Support vector machine *wo mochiita* chunk *dōtei. J. Natur. Lang. Process.* 9, 5 (2002), 3–21. In Japanese.
[24] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the EMNLP*. 230–237.
[25] John Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the ICML*. 282–289.
[26] Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the ACL*. 1064–1074.
[27] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn treebank. *Comput. Linguist.* 19, 2 (1993), 313–330.
[28] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network-based language model. In *Proceedings of Interspeech*, vol. 2. 1045–1048.
[29] Seung-Hoon Na. 2015. Conditional random fields for Korean morpheme segmentation and POS tagging. *ACM Trans. Asian Low-Res. Lang. Info. Process.* 14, 3 (2015), 10.
[30] Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. 2017. DyNet: The dynamic neural network toolkit. *arXiv:1701.03980* (2017).
[31] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proceedings of the ACL-HLT*. 529–533.
[32] Hideki Ogura, Hanae Koiso, Yumi Fujiike, Sayaka Miyauchi, Hikari Konishi, and Yutaka Hara. 2011. JC-D-10-05-01, and JC-D-10-05-02. Retrieved from http://pj.ninjal.ac.jp/corpus_center/bccwj/doc/report/JC-D-10-05-01.pdf; http://pj.ninjal.ac.jp/corpus_center/bccwj/doc/report/JC-D-10-05-02.pdf (in Japanese).
[33] John Okell and Anna Allott. 2001. *Burmese/Myanmar Dictionary of Grammatical Forms*. Routledge.
[34] Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the LREC*. 2089–2096.
[35] Lance A. Ramshaw and Mitchell P. Marcus. 1999. Text chunking using transformation-based learning. In *Natural Language Processing Using Very Large Corpora*. Springer, 157–176.
[36] Hammam Riza, Michael Purwoadi, Teduh Uliniansyah, Aw Ai Ti, Sharifah Mahani Aljunied, Luong Chi Mai, Vu Tat Thang, Nguyen Phuong Thai, Vichet Chea, Rapid Sun, Sethserey Sam, Sopheap Seng, Khin Mar Soe, Khin Thandar Nwet, Masao Utiyama, and Chenchen Ding. 2016. Introduction of the Asian language treebank. In *Proceedings of the O-COCOSDA*. 1–6.
[37] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1 (2014), 1929–1958.
[38] Karl Stratos and Michael Collins. 2015. Simple semi-supervised POS tagging. In *Proceedings of the NAACL-HLT*. 79–87.

[39] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the NIPS*. 3104–3112.
[40] Ann Taylor, Mitchell P. Marcus, and Beatrice Santorini. 2003. The Penn treebank: An overview. In *Treebanks*. Springer, 5–22.
[41] Thet Thet Zin, Khin Mar Soe, and Ni Lar Thein. 2011. Myanmar phrases translation model with morphological analysis for statistical Myanmar to English translation system. In *Proceedings of the PACLIC*. 130–139.
[42] Tin Htay Hlaing. 2012. Manually constructed context-free grammar for Myanmar syllable structure. In *Proceedings of the EACL*. 32–37.
[43] Sato Toshinori. 2015. Neologism dictionary based on the language resources on the Web for Mecab. Retrieved from https://github.com/neologd/mecab-ipadic-neologd.
[44] Kiyotaka Uchimoto, Qing Ma, Masaki Murata, Hiromi Ozaku, and Hitoshi Isahara. 2000. Named entity extraction based on a maximum entropy model and transformation rules. In *Proceedings of the ACL*. 326–335.
[45] Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2010. A unified character-based tagging framework for Chinese word segmentation. *ACM Trans. Asian Lang. Info. Process.* 9, 2 (2010), 5.

Received September 2017; revised April 2019; accepted April 2019