Towards Tokenization and Part-of-Speech Tagging for Khmer: Data and Discussion

HOUR KAINING, ASTREC, National Institute of Information and Communications Technology, Japan and Nara Institute of Science and Technology, Japan

CHENCHEN DING, MASAO UTIYAMA, and EIICHIRO SUMITA, ASTREC, National Institute of Information and Communications Technology, Japan

SETHSEREY SAM and SOPHEAP SENG, National Institute of Posts, Telecoms & ICT, Cambodia

KATSUHITO SUDOH and SATOSHI NAKAMURA, Nara Institute of Science and Technology, Japan

As a highly analytic language, Khmer has considerable ambiguities in tokenization and **part-of-speech (POS)** tagging processing. This topic is investigated in this study. Specifically, a 20,000-sentence Khmer corpus with manual tokenization and POS-tagging annotation is released after a series of work over the last 4 years. This is the largest morphologically annotated Khmer dataset as of 2020, when this article was prepared. Based on the annotated data, experiments were conducted to establish a comprehensive benchmark on the automatic processing of tokenization and POS-tagging for Khmer. Specifically, a support vector machine, a **conditional random field (CRF)**, a **long short-term memory (LSTM)**-based recurrent neural network, and an integrated LSTM-CRF model have been investigated and discussed. As a primary conclusion, processing at morpheme-level is satisfactory for the provided data. However, it is intrinsically difficult to identify further grammatical constituents of compounds or phrases because of the complex analytic features of the language. Syntactic annotation and automatic parsing for Khmer will be scheduled in the near future.

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

Additional Key Words and Phrases: Khmer, annotated data, tokenization, POS-tagging, machine learning

ACM Reference format:

Hour Kaing, Chenchen Ding, Masao Utiyama, Eiichiro Sumita, Sethserey Sam, Sopheap Seng, Katsuhito Sudoh, and Satoshi Nakamura. 2021. Towards Tokenization and Part-of-Speech Tagging for Khmer: Data and Discussion. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 20, 6, Article 104 (September 2021), 16 pages.

<https://doi.org/10.1145/3464378>

Chenchen Ding is Co-first author.

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. Authors' addresses: H. Kaing, ASTREC, National Institute of Information and Communications Technology, Japan, Nara Institute of Science and Technology, Japan; email: hour_kaing@nict.go.jp; C. Ding (corresponding author), M. Utiyama, and E. Sumita, ASTREC, National Institute of Information and Communications Technology, Japan; emails: {chenchen.ding, mutiyama, eiichiro.sumita}@nict.go.jp; S. Sam and S. Seng, National Institute of Posts, Telecoms & ICT, Cambodia; email: {sethserey.sam, sopheap.seng}@nptict.edu.kh; K. Sudoh and S. Nakamura, Nara Institute of Science and Technology, Japan; emails: {sudoh, s-nakamura}@is.naist.jp.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2375-4699/2021/09-ART104 \$15.00

<https://doi.org/10.1145/3464378>

1 INTRODUCTION

The Khmer language is the official language of the Kingdom of Cambodia. As a typical Austroasiatic language, Khmer is extremely analytic; syntactic information is overwhelmingly afforded by word order with abundant grammaticalization phenomena. Therefore, the resolution of morphological and syntactic ambiguities is contextually dependent. Figure 1 illustrates basic morphological features of the Khmer language.

Figure 1 also illustrates the Khmer scripts used to record the Khmer language. The specific writing system has a certain redundancy regarding the phonetic inventory, and the orthography is largely etymologically based. Because of the redundant writing system and highly analytic grammatical features, Khmer is not difficult to read, even without using separators to segment local meaning units in writing. Spaces are generally used as a comma or following the sentence-ending mark. Straightforwardly, the morphological processing of Khmer can be modeled as a cutting-and-tagging task in terms of engineering; however, the definition of tokens for cutting and the classification of the part-of-speech (POS) for tagging is intrinsically ambiguous, which makes the task challenging. Tokenization (or referred to as word segmentation) for Khmer has been studied by several researchers [2, 3, 15]. To accommodate the ambiguity of tokens, Chea et al. [3] defined several boundaries of tokens, that is, the boundary between words and boundaries between grammaticalized affixes and free morphemes. As the grammatical analysis of Khmer is heavily contextually dependent, a fundamental solution is to realize an ultimate joint processing for the grammatical category identification of different granularities. In this study, we focus on morphology to investigate joint processing for tokenization and POS-tagging. The ultimate joint processing involving syntactic parsing will be based on this work and scheduled in the near future.

This work is one component of the **Asian Language Treebank (ALT)** project [28], for which a Burmese version was the first completed treebank for a low-resource language [7, 8, 11]. Based on this previous work, we established a workflow to launch basic processing for a low-resource language, which was used to conduct the work on Khmer. As the primary contribution of this work, we prepared a systematically annotated 20,000-sentence Khmer corpus with tokenization and POS-tagging information using the nova scheme [9] from 2016. The data have been released under a CC BY-NC-SA license [10] for the research community.

Facilitated by the released data, we conducted experiments on four representative machine learning approaches to build a comprehensive benchmark for the automatic processing of Khmer at morphological level. Specifically, the approaches are (1) a point-wise classifier by a **support vector machine (SVM)**, which is a light-weight and fast solution, (2) **conditional random field (CRF)**, as a standard baseline for sequence labeling tasks, (3) an end-to-end solution by **recurrent neural network (RNN)** utilizing **long short-term memory (LSTM)** units, and (4) an integrated LSTM-CRF model as a state-of-the-art framework. We present analysis and discussions based on the experimental results to show the achievements and limitations of the data and approaches. As the Khmer language is the *most important member* in the Austroasiatic (Mon-Khmer) languages [12], this work can be a reference for the development of low-resource languages with similar features widely used in Southeast Asia.

The remainder of the article is organized as follows: In Section 2, we review related work, and in Section 3, we describe the data prepared in this study. In Sections 4 and 5, we present the experimental results and discussions, respectively. In Section 6, we conclude the article and state our future work.

2 RELATED WORK

Tokenization and POS-tagging are two classic tasks in **natural language processing (NLP)**, and are fundamental processing for many downstream NLP tasks and applications. This is particularly

Fig. 1. Formation of “*minister of defense*” in Khmer. The expression can be decomposed into five morphemes, which are in the upper rank. The first two morphemes, “*nation*” and “*official*”, are borrowed from Sanskrit/Pali. They form the expression “*minister*” in a head-final manner. The second half of “*ministry of defense*” is formed by Khmer-native morphemes in head-initial order, that is, the *department* that *protects* the *nation*. The entire expression is also formed head-initially, which is the predominant order in Khmer. Note that the expression “*ministry of defense*” can also be analyzed as the clause “*the department protects the nation*” as there is no grammatical declension nor conjugation paradigm in Khmer.

important for languages that apply a writing system without word separators, such as Chinese and Japanese. At an early stage, rule and dictionary-based matching approaches were applied, which required small resources. This is also a preliminary (or sole) approach for many understudied and low-resource languages. Unavoidable ambiguities revolved in tokenization and POS-tagging tasks can be further handled by statistical approaches once annotated data are provided. A classifier trained by SVM may provide a fast solution such as in Neubig and Mori [23]. As a more generalized framework for sequence labeling, CRF [19] is considered as a standard baseline. In recent years, neural networks, that is, models that use stacked multiple-layer nonlinear functions [5], have further generalized many typical NLP tasks under a sequence-to-sequence framework based on the methodology of end-to-end processing. The LSTM [13]-based RNN has been widely used as a feature evolved to capture long-range dependencies within complex sequences. A typical investigation is Yang et al. [31]. Furthermore, the features extracted by an LSTM-based RNN can be concatenated to a CRF interface for a better list-wise search [14, 20, 21, 27].

As reviewed, the technical background for tokenization and POS-tagging has been solidly established, and the techniques have been applied to many well-studied languages, such as Chen et al. [4], Ma and Hovy [21], and Zhao et al. [33] for Chinese word segmentation and Kudo et al. [18] for Japanese morphology analysis. When tokenization and POS-tagging tasks are both involved, the joint processing of the two tasks typically leads to good performance. In addition to Kudo et al. [18] for Japanese, a work on joint processing on Chinese is Kruengkrai et al. [17]. In our previous work on Burmese [7], we also discussed the effectiveness of joint processing on a low-resource language. Specifically, two difficulties exist for an understudied language: (1) data preparation and (2) practical engineering issues. Regarding (1), annotation requires native-speaker annotators. A linguistic background in the specific language is required in the design of the guidelines to construct explainable and consistent annotated data. Annotation processing is thus time-consuming. Regarding (2), a neural-network-based end-to-end approach can theoretically bridge the input and output directly without any further manual feature engineering. However, practical engineering issues still need to be examined, such as the hyperparameter selection, and more essentially, whether the quality and quantity of the available data are adequate to enable an end-to-end model to achieve decent performance.

A review of previous work on Khmer processing is as follows: Regarding tokenization, Seng et al. [29] and Bi and Taing [2] applied dictionary-based maximum matching; Huor et al. [15] further used corpus-based bi-gram language models; and Chea et al. [3] introduced the CRF. Regarding POS-tagging, Nou and Kameyama [26] proposed a template-based approach; Nou and Kameyama

[25] improved tagging for unknown words using language models. As to this study, data for joint tokenization and POS-tagging analysis are prepared and released, which are not only superior in terms of the scale to previous ones, but also containing annotation for more linguistic phenomena such as the grammaticalization and compounding. Based on the data, a comprehensive investigation on joint tokenization and POS-tagging processing for Khmer is conducted at the first time.

3 DATA

3.1 Overview

The Khmer corpus is a component of the ALT project, for which an overview is provided by Riza et al. [28]. Details are also described in previous works on Burmese [7, 8]. Briefly, the source data comprise 20,000 sentences collected from English *Wikinews*, and the English sentences are translated into different Asian low-resource languages by professional translators.¹ The ultimate aim is to build a syntactically annotated treebank for each language. The Khmer data released in the present work are a stable morphologically annotated version. A joint tokenization and POS-tagging scheme nova Ding et al. [9] is applied and the annotation guidelines have been released.² In this section, a brief review of the manual annotation process is provided in Section 3.2, important issues related to the annotation is described in Section 3.3, and overall statistics with typical annotation examples are presented in Section 3.4.

3.2 Timeline

Compared with our previous work on Burmese, where around thirty annotators were involved [7], the team for Khmer annotation comprised only around ten people, and at most times, fewer than five people. Except the co-first author, the annotators are Khmer native-speakers. All the team members are researchers and students proficient in NLP. It takes around 4 years to prepare the released data of this work. The following is a brief retrospect organized in four stages:

- (1) In 2016, the annotation guidelines were edited by the first and co-first authors. The annotation was preliminary based on in-house data provided by Chea et al. [3]. Automatic tools for tokenization and POS-tagging were trained and applied to raw Khmer data. One native-speaker annotator (the first author) began to perform manual checking on the automatic processed data from the second half of 2016.
- (2) In 2017, seven further native-speaker annotators were involved in updating the data according to the guidelines. At the end of 2017, the data had been manually tokenized and annotated using the basic/auxiliary tags of the nova scheme.
- (3) In 2018, the annotation was extended with the modified nova tags. The supplementary material of the guidelines was edited. The work in 2018 was conducted by the first and co-first authors with two other native-speaker annotators.
- (4) In 2019, the first and co-first authors, with another native-speaker annotator, began the syntactic annotation, along which the tokenization and POS-tagging were further polished.

3.3 Important Issue

For tokenization, two types of tokens, short ones and long ones, were designed to accommodate the *morpheme* and *word* units, respectively. In the annotation, spaces were inserted to separate short tokens; long tokens were bracket-wrapped short token sequences. Smaller than a short token,

¹Stylistically, modern colloquial styles used for daily and business communications were required in the translation, as the project focuses on developing NLP engineering techniques for contemporary societies.

²Annotation guidelines: <http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/Khmer-annotation-guideline.pdf> and <http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/Khmer-annotation-guideline-supplementary.pdf>.

Table 1. nova Tags Used in Annotation

basic tags	
n	general <u>n</u> ouns, can be subjects or objects of tokens tagged by v
v	general <u>v</u> erbs, can take tokens tagged by n as arguments
a	general <u>a</u> djectives, can directly describe or modify tokens tagged by n
o	<u>o</u> ther modifications or complements, for which n, v, and a tags are not applicable
auxiliary tags	
1	general numbers
.	general punctuation marks
+	a catch-all category, for tokens with weak syntactic roles, such as interjections or fillers
modified tags	
n-	function n-tagged tokens, general pronouns
v-	function v-tagged tokens, adpositions or particles from grammaticalized verbs
a-	function a-tagged tokens, general articles or determiners
o-	a catch-all category for fuction tokens

the basic atom in Khmer scripts is referred to as a *writing unit*. It comprises one or more staked consonant characters, with one or more optional modifying diacritics. Phonetically, a writing unit is not strictly consistent with the syllable structure [6]; however, it is the minimal unbreakable atom in the writing system that is suitable for textual data processing.³

In practice, the short and long tokens were realized by *aggressive* and *conservative* segmentation according to the native-speaker annotators' common sense. Specifically, the annotators were instructed to segment the data into meaningful units as small as possible to generate the short tokens, where more segmented instances are preferable if inconsistency arose among annotators. Several conventional Khmer dictionaries are referred to in the annotation for the identification of long tokens. Generally, a sequence composed by one or more short tokens is identified as a long token if listed as an entry in dictionaries. For those expressions that cannot be covered by dictionaries, the annotation is largely dependent on the common sense, where extremely complex long tokens might be generated due to compounding. In practice, annotators were suggested to restrict a long token composed no more than *four* short tokens to avoid long tokens that are too complex. The annotation related to foreign names and loanwords was based on the original languages. Each single word in the original languages was treated as a short token. Proper nouns and phrases composed of multiple words were annotated as one long token.

As to the POS-tagging, Table 1 lists the applied tags from the annotation guidelines. Each short token is tagged with one POS tag, and each long token composed by multiple short tokens is tagged with a second-layer POS tag outside a pair of wrapping brackets. In the released data, the basic/auxiliary/modified tags are treated equally. An annotation example is provided in Section 3.4 for illustration. The released guidelines provided basic principles of the use of the tags for annotators. In practice, the same approach of cross-checking in the previous work of Burmese [7] was applied to improve the consistency among annotators.

3.4 Statistics and Example

Table 2 lists the statistics of the annotated data. The division of the training/development/test sets was performed according to the unified setting under the ALT project. Figure 2 shows an example

³Compared with Burmese processing, where the syllable is used as the natural atom [7], Khmer has a more complex syllable structure and less clear boundaries between syllables in writing.

Table 2. Statistics for the Released ALT Khmer Data

dataset	#writing unit	#token		#sentence
		short	long	
training	1,245,497	643,785	478,964	18,088
development	68,318	35,221	26,207	1,000
test	69,616	36,007	26,551	1,008
total	1,303,431	715,013	531,722	20,106
average writing unit(s)	1	1.9	2.6	68.8

from the released data. The 19 indexed tokens are short tokens. In terms of long tokens, the bracketed **4 5**, **9 10**, **13 14 15**, and **16 17** are considered as single long tokens. In the case of the nova annotation of Burmese [7], the *v-* tag is not used; however, this tag is frequently used for Khmer. This is because Khmer has a huge grammaticalized phenomena that many functional particles are derived from basic verbal morphemes [12]. The tokens **3**, **6**, and **18** were originally the verbal morphemes for “to say”, “to be able to”, and “to get”, respectively. **3** has become a conjunction for direct/indirect speech, and **6** has become like an auxiliary verb. The meaning of **18** has become vague, and it appears as a sentence-ending particle with a collocation of **6**. All these tokens were annotated by *v* at stage (2) mentioned in Section 3.2 and then further modified into *v-* at stage (3). Compounding is abundant in Khmer, as the constituents of **4 5** and **13 14 15** show. Note that both of them have a purely head-initial structure, as described in Figure 1. The compound of **9 10** is another common structure in which morphemes with identical/similar meanings are duplicated for emphasis. The constituent of **16 17** is a numerical attribute that modifies **13 14 15**; note that **13** and **17** are identical morphemes, with the original meaning of “grain/seed”. In the compound of **13 14 15**, it has a more specifically derived meaning of “bullet”, and as **17**, it is grammaticalized as a counter for seed-like objects. From this annotation example, it can be observed that POS conversion (i.e., zero derivation) is common in Khmer, which may lead to difficulties in automatic processing.

4 EXPERIMENT

4.1 Setting

As mentioned in Section 3.4, the writing units in the Khmer scripts were used as the atom tokens. Further tokenization and POS-tagging processing for short and long tokens were thus unified into a sequence labeling task for these writing units. The IBES scheme to represent inside a token, beginning of a token, end of a token, and single unit, was used by default. Simplified schemes of IE and IB were also compared in experiments. As the configuration of processing, cascaded and jointed manners were compared. The former one is to tokenize first and then to tag the tokens using two separate models; the later one is to generate the token boundary with the POS information simultaneously using one model.

For a comprehensive investigation, four representative machine learning approaches are experimented with the following settings: The results by each approach are described in the following subsections, respectively:

- A linear SVM wrapped by Kytea⁴ [24] was used as a preliminary fast solution. The SVM performs point-wise predication for each writing unit with a sliding window of contextual

⁴<http://www.phontron.com/kytea>.

Fig. 2. Tokenized and POS-tagged Khmer sentence. English glosses for Khmer short tokens and bracketed long tokens are attached at the top of the figure. The boundaries between writing units within each short token are illustrated by vertical broken lines. The original English sentence is “Officials say the rods could be able to provide enough plutonium to make two nuclear bombs.”

Fig. 3. Configuration of the network of LSTM-based RNN used in the experiment. The numbers represent the vector dimensions.

Fig. 4. Configuration of the network of LSTM-CRF used in the experiment. The numbers represent the vector dimensions.

features. Bi-, tri-, and 4-gram features were experimented. As a tradeoff of the fastness, the point-wise estimation is intrinsically weak in capturing the sequential information. Only the complete IBES scheme was applied with this method to encode more contextual information.

- A CRF model implemented by the CRF++ toolkit⁵ was set as a standard sequence labeling baseline. The settings and feature template followed the previous work [7].
- An LSTM-based RNN was implemented using DyNet⁶ [22]. The network was essentially configured according to Ding et al. [7], whereas the dimensions of layers were enlarged to achieve better performance. A diagram of the network configuration is shown in Figure 3. The parameters were trained using Adam [16] with the default setting in DyNet.⁷ The model ensemble was also conducted, up to 100 models, as large-scale model-ensemble was found gradually increased the performance.
- An LSTM-CRF was implemented using NCRF++⁸ [32]. The network configuration followed that in Ma and Hovy [21] with the dimension of layers adjusted to fit our task. A diagram of the network configuration is shown in Figure 4. The stochastic gradient descent method is used to optimized the model with the default setting in NCRF++.⁹ The loss of each batch was averaged and the model was trained with 100 iterations. The model ensemble was also conducted up to 100 models.

For the evaluation, the F-score, which is the harmonic mean of the precision and recall, was used. For tokenization, precision is the ratio between the number of correct segmented tokens and the total number of tokens obtained by automatic processing, and recall is the ratio between the correct segmented tokens and the total number of manually segmented tokens in the reference. For POS-tagging, the F-score was calculated jointly on correct tokens with correct POS tags.

⁵<https://taku910.github.io/crfpp/>.

⁶<https://github.com/clab/dynet>.

⁷Learning rate $\alpha = 10^{-3}$, moving average for the mean/variance $\beta_1 = 0.9$, $\beta_2 = 0.999$, and bias $v = 10^{-8}$.

⁸<https://github.com/jiesutd/NCRFpp>.

⁹10 for batch size and 0 for momentum. Initial learning rate $\eta_0 = 0.015$ and decay rate $\rho = 0.05$.

Table 3. F-core for Tokenization and POS-Tagging Using SVM Based on Writing Units

	cascaded				joint			
	token		POS		token		POS	
	short	long	short	long	short	long	short	long
2-GRAM/IBES	0.971	0.870	0.951	0.851	0.972	0.883	0.946	0.851
3-GRAM/IBES	0.977	0.880	0.953	0.858	0.976	0.887	0.951	0.860
4-GRAM/IBES	0.976	0.882	0.952	0.855	0.976	0.888	0.951	0.862

Table 4. F-Score for Tokenization and POS-Tagging Using CRFs Based on Writing Units

	cascaded				joint			
	token		POS		token		POS	
	short	long	short	long	short	long	short	long
2-GRAM/IBES	0.980	0.875	0.959	0.855	0.980	0.886	0.959	0.873
3-GRAM/IBES	0.982	0.896	0.961	0.876	0.980	0.896	0.962	0.883
4-GRAM/IBES	0.982	0.899	0.960	0.878	0.979	0.898	0.960	0.884
2-GRAM/IB	0.969	0.837	0.948	0.818	0.972	0.870	0.951	0.856
3-GRAM/IB	0.977	0.885	0.956	0.865	0.976	0.886	0.958	0.872
4-GRAM/IB	0.977	0.888	0.956	0.868	0.975	0.888	0.957	0.874
2-GRAM/IE	0.971	0.851	0.950	0.831	0.974	0.879	0.953	0.865
3-GRAM/IE	0.976	0.889	0.955	0.869	0.976	0.892	0.957	0.878
4-GRAM/IE	0.976	0.890	0.956	0.870	0.975	0.891	0.957	0.877

4.2 SVM

Table 3 lists the F-score of tokenization and POS-tagging using the SVM. As a fast and preliminary solution, several overall tendencies can be observed:

- The gains brought by tri-gram features were more obvious than those brought by 4-gram features. In cascaded processing and on the processing of short tokens, the 4-gram features did not bring substantial improvements. In joint processing and on the processing of long tokens, the 4-gram features provided limited improvements.
- The joint processing only boosted the performance on long tokens up to the POS-tagging, where the longer N -gram features gave consistent increasing gains.

These tendencies suggest that the joint processing combined with complex features can help difficult tasks related to long tokens and POS-tagging. As the capability of the point-wise processing is limited. We move on to the results of CRF in the next subsection as a formal baseline.

4.3 CRF

Table 4 lists the F-score of tokenization and POS-tagging using the CRF. The overall performance is much better than that by the SVM, while similar tendencies can be observed. As a primary finding on the tagging scheme, the IBES scheme was superior to the IB or IE schemes under all settings. This is reasonable and in accordance with previous studies in which the IBES scheme coded more boundary information.¹⁰

¹⁰As an interesting side finding, the IE scheme was slightly better than the IB scheme, particularly for long tokens. In Ding et al. [7], the authors mentioned that the IB scheme typically outperformed the IE scheme for head-final languages. Here the reverse was observed on Khmer, which is a typical head-initial language.

Table 5. F-Score for Tokenization and POS-Tagging Using LSTM-based RNN Based on Writing Units

	cascaded				joint			
	token		POS		token		POS	
	short	long	short	long	short	long	short	long
MINIMUM @100	0.975	0.883	0.955	0.866	0.977	0.879	0.954	0.856
MAXIMUM @100	0.979	0.891	0.961	0.878	0.979	0.892	0.960	0.871
10-ENSEMBLE	0.981	0.903	0.965	0.890	0.983	0.903	0.966	0.888
20-ENSEMBLE	0.982	0.902	0.966	0.890	0.983	0.906	0.967	0.891
50-ENSEMBLE	0.982	0.904	0.966	0.891	0.983	0.906	0.967	0.891
100-ENSEMBLE	0.982	0.904	0.966	0.891	0.983	0.906	0.967	0.892

Table 2 shows that the short tokens comprised fewer than two writing units, on average. Therefore, tri-gram features were adequate for short token identification. Generally, the performance for short token processing was quite high, even using cascaded processing, and joint processing did not result in substantial gains (from 0.961 to 0.962; no statistical significance at the 5% level). This suggests that short tokens, that is, morpheme-level units, were relatively consistent in the manual annotation, as well as in the automatic processing. The performance on long tokens obviously decreased. Using joint processing, there was a limited gain from 0.878 to 0.884 in terms of POS-tagging, but no difference for tokenization (0.899 and 0.898; no statistical significance at the 5% level). In practical annotation, long tokens accommodated complex and diverse phenomena, such as compounding and grammaticalization, which led to difficulties in automatic processing. The reported results on long tokens in Table 4 were generated directly from writing units. When we introduced an extra cascaded step to generate short tokens first and then combine them to long tokens, performance dropped to around 0.81. Therefore, even though the gain that resulted from joint processing was limited, it was indispensable for long token processing.

4.4 RNN

Table 5 presents the results from 10- to 100-model ensembles. The performance of the best/worst model among the 100 models are also presented for comparison. The ensemble made certain gains, which were more obvious for long tokens than short tokens, and more obvious for POS-tagging processing than tokenization processing. Experiments were conducted on both cascaded and joint processing using the RNN, and joint processing always outperformed cascaded processing. This is in accordance with the end-to-end methodology of neural network-based approaches. The best performance for short tokens was 0.983 for tokenization and 0.967 for POS-tagging. The improvement for tokenization was limited compared with the CRFs best results of 0.982; the improvement for POS-tagging was slightly larger: from the CRFs 0.962 to 0.967. For long tokens, the best performance for tokenization was 0.906, and for POS-tagging was 0.892. Comparing the respective best results of 0.899 and 0.884 using the CRF, the gains were more obvious than those for short tokens.

Based on the experimental results, the RNN can be considered as a more reasonable method than the CRF for Khmer processing when joint processing is required, for example, to identify grammatical constituents larger than morphemes, or further, to obtain the grammatical categories for them. Conversely, the task of only identifying the boundary at morpheme-level is not difficult; and the CRF will be a prompt option.

Table 6. F-Score for Tokenization and POS-Tagging using LSTM-CRF based on Writing Units

	cascaded				joint			
	token		POS		token		POS	
	short	long	short	long	short	long	short	long
MINIMUM @100	0.978	0.887	0.962	0.875	0.979	0.892	0.961	0.878
MAXIMUM @100	0.981	0.894	0.965	0.881	0.982	0.898	0.964	0.885
10-ENSEMBLE	0.982	0.895	0.965	0.884	0.982	0.898	0.966	0.885
20-ENSEMBLE	0.982	0.897	0.966	0.885	0.982	0.899	0.966	0.886
50-ENSEMBLE	0.982	0.898	0.967	0.886	0.983	0.900	0.967	0.888
100-ENSEMBLE	0.982	0.897	0.966	0.885	0.983	0.900	0.966	0.888

4.5 LSTM-CRF

The results by LSTM-CRF are presented in Table 6. From the results of the best/worst model, LSTM-CRF generally provided better and more stable performance than that of the plain RNN in terms of single models. However, multi-model ensemble was more effective on the models trained by the RNN than those by the LSTM-CRF from the results of large-scale model ensemble. On short tokens, the RNN and LSTM-CRF gave very close performance in joint processing after ensemble, i.e., 0.983 vs. 0.983 on tokenization and 0.967 vs. 0.967 on POS-tagging. On long tokens, the ensemble improved the best RNN model from 0.892 to 0.906 (+0.008), while only from 0.898 to 0.900 (+0.002) for LSTM-CRF, in terms of tokenization. The differences were more obvious on the POS-tagging for long tokens, where ensemble boosted the RNN from 0.871 to 0.892 (+0.021) while only 0.885 to 0.888 (+0.003) for LSTM-CRF. Consequently, the best ensembled results by RNN outperformed those by LSTM-CRF with statistical significance at the 1% level on the long tokens. As the ensemble theory [30] suggests that more diverged models tend to yield better performance, it can be considered that those RNN models suffered a more unstable training while benefited more from the ensemble, compared with the LSTM-CRF where single models were stronger but with less diversities.

5 DISCUSSION

5.1 Short Token

As the RNN and LSTM-CRF had nearly identical performance on short tokens, here the comparison between CRF baseline and RNN is focused. Typical tokenization errors for short tokens made by both the CRF and RNN are listed in Figure 5. Some of the errors were caused by the intrinsic analytic features of Khmer, for example, the word for “train” comprises “vehicle” and “fire”; “queen” has a prefix for emphasis; and the duplicated expression of the preposition “in”. Strictly, these types of over-segmented¹¹ results using automatic processing were more consistent with our tokenization principles, compared with the manual segmentation used for reference. Therefore, these errors were not serious but just a reflection of the difficulty of manual annotation. The main problem in tokenization was around non-native proper nouns, for example, “Transnistria” and “Jerusalem”. The difference in performance between the RNN and CRF was mainly caused by such errors for proper nouns, where the RNN provided better identification of boundaries using contextual information. Introducing a large-scale dictionary should be an efficient solution to this issue.

¹¹If the automatic processing generates a more fragmentary segmentation compared with the manual annotation, it is referred to as “over-segmented”, and a less fragmentary segmentation is referred to as “under-segmented”.

manually segmented token		automatically segmented token	
រថភ្លើង	train	រថ	vehicle
ម្ចាស់ក្សត្រី	queen	ភ្លើង	fire
នៅក្នុង	in	ម្ចាស់	owner
ព្រំនីស្ទីន	Transnistria	ក្សត្រី	queen
ជួរសាលីម	Jerusalem	នៅ	at
		ក្នុង	in
		ព្រំនីស្ទី	*transni-
		ស្ទីន	*-stria
		ជួរ	*jeru-
		សាលីម	*-salem

Fig. 5. Typical errors for short tokens on the test set. (* represents meaningless fragments).

token	នៅ to be in		ជា to be as		នៅ to go to		ឡើង to go up		អ្នក person, you		មួយ one, single	
tag	v-	v	v	v-	v-	v	v-	v	n	n-	1	a
(%)	70.8	29.2	82.5	17.3	55.2	44.8	58.3	41.7	74.4	25.6	67.6	32.4

Fig. 6. Distribution of manually annotated POS tags for short tokens difficult to process automatically.

In joint processing, the errors were mostly related to POS-tagging rather than tokenization. Typical words with confusing POS tags are listed in Figure 6, with the tag distribution from manual annotation. The most common type of error was around grammaticalized verbal tokens, which may act like prepositions, for example, the words for “to be in”, “to be as”,¹² and “to go to”,¹³ or act like particles, for example, the word for “to go up”. The RNN generally outperformed both CRF for these verbal tokens because manual annotation is quite contextually dependent and the RNN can model long-distance contextual information better than the CRF. Other common errors were around some intrinsic ambiguities in POS. For example, pronouns in Khmer are generally derived from various nouns. The common second person pronoun in Khmer is actually from the noun for “a person”, which is a more frequent meaning in the data. Distinguishing between its role as a pronoun and as a common noun is difficult no matter the specific methods, as the syntactic roles are generally the same and there is no verbal conjugation in Khmer to help disambiguate them. The numerical word for “one” can also be used as an adjective for “single” depending on the context. In this case, disambiguation is also difficult.

5.2 Long Token

As long tokens cover a large range of complex phenomena, a lexicalized statistic results in a long tail of singletons. Table 7 shows frequent patterns of POS-tagging errors for long tokens, in addition to the distribution of the number of tokens in these errors. The under-segmentation tendency of the CRF and over-segmentation tendency of the RNN are obvious. The LSTM-CRF has a mixed tendency of CRF and RNN, where the over-segmentation is similar to RNN but the patterns of POS errors are like those of CRF.

¹²A copula and somehow a grammaticalized instrumental case marker.

¹³Somehow, a grammaticalized dative case marker.

Table 7. Error Distribution for Long Tokens at the POS-tagging Level and for the Number of tokens

	CRF 4-GRAM		RNN 100-ENSEMBLE		LSTM-CRF 50-ENSEMBLE	
POS pattern	n n -> n	18.3%	n -> n n	19.9%	n n -> n	16.5%
	n -> n n	13.2%	n n -> n	12.8%	n -> n n	16.3%
	n v -> n	2.9%	n v -> n	3.7%	n v -> n	3.9%
token number	2 -> 1	32.7%	1 -> 2	37.2%	1 -> 2	32.5%
	1 -> 2	26.5%	2 -> 1	23.9%	2 -> 1	29.9%
	1 -> 1	15.0%	1 -> 1	17.2%	1 -> 1	13.7%

Top-3 frequent patterns are listed. For each item, the left side of the arrow (->) represents the manual annotation and right side represents the result using automatic joint processing.

manually segmented token		automatically segmented token	
ប្រាក់	money	ឈ្នួល	salary
ប៉ូល	pole	ខាងជើង	north
ក្រុមអូស្ត្រាលី	Australian team	ក្រុម	team
ការប្រកួតស៊េរី	series match	អូស្ត្រាលី	Australia
		ការប្រកួត	match
		ស៊េរី	series

Fig. 7. Examples of under-segmented nominal expressions using the CRF (top) and over-segmentation nominal expressions using the RNN (bottom) compared with manual annotation.

Typical under-/over-segmentation examples are listed in Figure 7. The example of “salary” is similar to the example of “queen” in Figure 5; the expression is segmented into two tokens in the manual annotation. As such an inconsistency lies on the borderline of the definition of the word, it is not so serious in practice. However, the other three examples reveal a complex problem related to the annotation of expressions that involve borrowed words. Here the Khmer expressions for “pole”, “Australia”, and “series” are directly borrowed from English, and the expressions for “north”, “team”, and “match” are Khmer natives. Confusion arises in such cases. For the expression for “the North Pole”, manual annotation was performed according to the principle for detailed segmentation. However, the Khmer morpheme for “team” has been grammaticalized, and the “series” is not a stable borrowed word. Annotators may hesitate when tokenizing such hybrid expressions.

5.3 Data Size and Further Improvement

Figure 8 shows the effect of the data size on the performance of joint processing. The rightmost points are the above-reported results using the full training data, and the performances using half, one-quarter, and one-eighth of the training data are plotted. The results show that the differences between the RNN and the LSTM-CRF are gradually reduced when more training data were provided. However, the RNN always outperformed the CRF, regardless of the size of the dataset.

This suggests that performance can be further boosted by increasing the training data. Especially on long tokens, the LSTM-CRF can be expected to outperform the RNN. However, for the data provided in this study, we consider that integrating the resources of proper nouns is the most practical direction to improve automatic processing. Regarding the POS-tagging, there are intrinsic ambiguities for certain morphemes because of the highly analytic features of Khmer. A solution

Fig. 8. F-score for joint processing on short (left) and long (right) tokens on different training data sizes. (x-axes for the number of tokens, in logarithmic scale).

may rely on a more informative POS-tag set than the nova scheme used in this study. This requires a more insightful, linguistically oriented investigation of the Khmer language.

6 CONCLUSION AND FUTURE WORK

In this study, we primarily proposed a 20,000-sentence tokenized and POS-tagged Khmer corpus. The manual annotation processing started in 2016, and the corpus was the largest morphological annotated Khmer dataset when this article was written in 2020. Based on the annotated data, experiments were conducted for automatic tokenization and POS-tagging using four standard approaches of SVM, CRF, LSTM-RNN, and LSTM-CRF. Experiment-based analysis showed that automatic processing up to morpheme-level was satisfactory, but for larger constituents, for example, complex compounds and phrases, joint processing was required.

The syntactic annotation on the identical corpus is ongoing to complete a final treebank of Khmer. We plan to conduct experiments on the ultimate joint processing of tokenization, POS-tagging, and syntactic parsing when we reach a stable version of the treebank. Additionally, as the experiments showed, a proper noun dictionary is necessary. In our previous work, we collected a dictionary of borrowed words in Burmese [1] after completing the Burmese treebank. This is also a direction for our future work on Khmer.

Broadly, the discussed linguistic phenomena of the Khmer language, that is, the head-initial and analytic features, are prevalent for many languages in Southeast Asia. The investigation in this study can also contribute to the development of these languages, most of which are still low-resource and understudied in the NLP field.

ACKNOWLEDGMENTS

We thank all the translators who provided the raw Khmer ALT data. We thank the annotators mentioned in Section 3.2 for the annotation. They are Ms. Bolin Prak, Ms. Sophorsvotey Sreang, Mr. Kimleng Meng, Mr. Solay Heng, Mr. Thothaivorn Sory, Mr. Krisna Mony Or, and Mr. Pharo Set, for stage (2), Mr. Kamthong Ley and Mr. Vanna Chuon, for stage (3), and Mr. Porchourng Chuor for stage (4).

REFERENCES

- [1] Aye Myat Mon, Chenchen Ding, Hour Kaing, Khin Mar Soe, Masao Utiyama, and Eiichiro Sumita. 2020. A Myanmar (Burmese)-English named entity transliteration dictionary. In *Proceedings of LREC*. 2980–2983.
- [2] Narin Bi and Nguonly Taing. 2014. Khmer word segmentation based on bi-directional maximal matching for plaintext and microsoft word document. In *Proceedings of APSIPA*. 1–9.
- [3] Vichet Chea, Ye Kyaw Thu, Chenchen Ding, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2015. Khmer word segmentation using conditional random fields. In *Proceedings of Khmer NLP*. 62–69.

- [4] Xinchu Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuan-Jing Huang. 2015. Long short-term memory neural networks for Chinese word segmentation. In *Proceedings of EMNLP*. 1197–1206.
- [5] 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.
- [6] Chenchen Ding, Vichet Chea, Masao Utiyama, Eiichiro Sumita, Sethserey Sam, and Sopheap Seng. 2017. Statistical khmer name romanization. In *Proceedings of PACLING*. 179–190.
- [7] 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. Info. Process.* 19, 1 (2019), 5.
- [8] Chenchen Ding, Sann Su Su Yee, Win Pa Pa, Khin Mar Soe, Masao Utiyama, and Eiichiro Sumita. 2020. A Burmese (Myanmar) treebank: Guideline and analysis. *ACM Trans. Asian Low-Resour. Lang. Info. Process.* 19, 3 (2020), 40.
- [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, Masao Utiyama, and Eiichiro Sumita. 2020. Tokenized and POS-tagged Khmer data of the Asian language treebank project. <https://doi.org/10.5281/zenodo.3937914>
- [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), 22.
- [12] Madeline Elizabeth Ehrman and Kem Sos. 1972. *Contemporary Cambodian: Grammatical Sketch*.
- [13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
- [14] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).
- [15] Chea Sok Huor, Top Rithy, Ros Pich Hemy, Vann Navy, Chin Chanthirith, and Chhoeun Tola. 2004. Word bigram vs orthographic syllable bigram in Khmer word segmentation. *PAN Localization Working Papers* (2004).
- [16] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.
- [17] Canasai Kruengkrai, Kiyotaka Uchimoto, Junichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009. An error-driven word-character hybrid model for joint Chinese word segmentation and POS tagging. In *Proceedings of ACL-AFNL*. 513–521.
- [18] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of EMNLP*. 230–237.
- [19] John Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*. 282–289.
- [20] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*. 260–270.
- [21] Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of ACL*. 1064–1074.
- [22] Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqi, 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 preprint arXiv:1701.03980* (2017).
- [23] Graham Neubig and Shinsuke Mori. 2010. Word-based partial annotation for efficient corpus construction. In *Proceedings of LREC*.
- [24] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proc. of ACL*. 529–533.
- [25] Chenda Nou and Wataru Kameyama. 2007. Hybrid approach for Khmer unknown word POS guessing. In *Proceedings of IRI*. 215–220.
- [26] Chenda Nou and Wataru Kameyama. 2007. Transformation-based Khmer part-of-speech tagger. In *Proceedings of ICAI*. 581–587.
- [27] Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of ACL*. 1756–1765.
- [28] Hammam Riza, Michael Purwadi, Gunarso, Teduh Uliniansyah, Aw Ai Ti, Sharifah Mahani Aljunied, Luong Chi Mai, Vu Tat Thang, Nguyen Phuong Thai, Rapid Sun, Vichet Chea, 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 O-COCOSDA*. 1–6.
- [29] Sopheap Seng, Sethserey Sam, Laurent Besacier, Brigitte Bigi, and Eric Castelli. 2008. First broadcast news transcription system for Khmer language. In *Proceedings of LREC*. 2658–2661.

- [30] Peter Sollich and Anders Krogh. 1995. Learning with ensembles: How overfitting can be useful. In *Proceedings of NIPS*. 190–196.
- [31] Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of COLING*. 3879–3889.
- [32] Jie Yang and Yue Zhang. 2018. NCRF++: An open-source neural sequence labeling toolkit. In *Proceedings of ACL, System Demonstrations*. 74–79.
- [33] 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), 1–32.

Received July 2020; revised February 2021; accepted April 2021