Doctoral Dissertation

Towards Morphological And Syntactic Analyses For The Khmer Language

Hour Kaing

Program of Information Science and Engineering
Graduate School of Science and Technology
Nara Institute of Science and Technology

Supervisor: Professor Satoshi Nakamura
Augmented Human Communication Lab.
(Division of Information Science)

Submitted on March 17, 2022

A Doctoral Dissertation
submitted to Graduate School of Science and Technology,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of Engineering

Hour Kaing

Thesis Committee:

Supervisor Satoshi Nakamura
(Professor, Division of Information Science)
Taro Watanabe
(Professor, Division of Information Science)
Katsuhito Sudoh
(Associate Professor, Division of Information Science)
Masao Utiyama
(Executive Researcher, NICT)

Towards Morphological And Syntactic Analyses For The Khmer Language*

Hour Kaing

Abstract

Language is a structural system that humans use for communication. Humans analyze structures of sentences to understand meanings and to generate grammatical sentences. In natural language processing (NLP), the same structural analysis is necessary to enhance computers' understanding and generation of natural language sentences. Its usefulness has been demonstrated for many downstream systems, e.g., language model, machine translation, text summarization. Even though the structural analysis for computers has been studied for a long time, many languages have not yet been investigated because of their resource scarcity and a requirement of language-specific knowledge. This problem is crucial, particularly in this multilingual NLP era. This thesis concerns morphological and syntactic analyses for a low-resource language—Khmer. We organize our contributions into two parts.

Part one is about morphological analysis, the most fundamental task in NLP. The Khmer language is highly analytic and has no word boundary indicators. Previously, tokenization and part-of-speech (POS) tagging have been studied for words and considered two separate tasks. For the Khmer language, the definition of tokens and the classification of the POS is intrinsically ambiguous, which makes the tasks challenging. This thesis focuses on morphology to investigate joint processing for tokenization and POS-tagging. Our contributions are a preparation of the largest tokenized and POS-tagged corpus that contains annotation for more linguistic phenomena than a previous corpus, and an investigation of the automatic processing of tokenization and POS tagging on this corpus using various

*Doctoral Dissertation, Graduate School of Science and Technology, Nara Institute of Science and Technology, March 17, 2022.

state-of-the-art approaches. We present analyses and discussions based on our experimental results to show the achievements and limitations of the data and approaches.

Part two is about syntactic analysis for Khmer. We focus on constituency parsing, which is another core NLP task. Because the development of constituency treebank is costly, we investigate cross-lingual transfer learning techniques for constituency parsing in this thesis. We conduct two experiments for many diverse languages in addition to Khmer. The first experiment discusses the single-source transfer performance of a POS-based delexicalized model and the effectiveness of various source-language selection techniques. Then, the second experiment examines a multi-source transfer using a pretrained multilingual language model as a cross-lingual channel. Because a multilingual constituency treebank consists of diverse structures and label sets, we propose a treebank preprocessing step and typological features integration with smooth sampling and dropout to improve the cross-lingual performance.

Keywords:

Natural Language Processing, Morphology, Tokenization, Part-of-Speech, Constituency Parsing, Under-Studied, Low-Resource, Cross-Lingual Transfer

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Language is a structural system that humans use to communicate and express thoughts. Humans use language as a set of constraints that connect symbols to represent meanings. In linguistics, the term syntax or grammar is used to describe such structural constraints on how speakers or writers compose clauses, phrases, and words. Grammars of a natural language are used as a convention to describe the grammatical correctness of the language. For language learners, the grammars are also used as an explicit instruction to analyze and construct a sentence of the language.

Natural language processing (NLP) is a study hoping that computers can understand and generate natural language texts like humans. It allows the interaction between humans–and–computers or humans–and–humans (e.g., across languages) to be more natural and efficient than before. For many decades, many tasks have been created for various purposes. Some tasks have their direct real-world applications, e.g., text summarization, grammatical error correction, machine translation, question answering. Some other tasks are sub-tasks that support the larger tasks, e.g., morphological analysis, syntactic analysis, semantic analysis. The sub-tasks are extremely important for NLP, especially those that analyze the structures of the texts, because a natural language text in computers is just a sequence of characters. Those structural analyses allow the computers to capture syntactically and semantically useful information from the plain texts or to process language more naturally.

The morphological and syntactic analyses are considered as the two fundamental steps in NLP as humans generally analyze structures of words and sentences. The morphological analysis describes how a word is formed and simultaneously looks at the linguistic categories of its elements (morphemes) as well as itself. Words in English are easily detected by the computers because spaces are used to separate words. However, it may not be the case for other languages, e.g.,

Chinese, Japanese, Khmer, because spaces are used for various purposes depending on the nature of each language. Subsequently, the morphological analysis of those languages has been modeled jointly with word tokenization as a cutting-and-tagging task in terms of engineering [1, 2]. For the other NLP tasks, the tokenization is required as the earliest step, and the POS information is extremely valuable for the improvement, e.g., machine translation [3–7], question answering [8], dialog system [9]. In addition to the analysis of words, the syntactic analysis, which is a process to analyze the structure of a sentence, is also regarded as a core NLP task because sentences meanings are more than the sum of the meanings of their words rather the internal structures of the sentences. To the larger tasks, the syntactic structures can be beneficial, e.g., for language model [10], machine translation [5, 7, 11–15], text summarization system [16].

The morphological and syntactic analyses are still under-studied and under-developed for many languages. In the traditional NLP, these analyzers were developed based on a collection of predefined human rules, which is hard to be developed and not applicable for texts in the real world. Until the computers turn powerful, data-driven approaches became more practical solutions for almost every NLP task where those rules can be learned from a large amount of human-annotated data. Remarkably, data resources are gradually available for more and more languages by the contributors from various nations, e.g., more than 100 languages in the Universal Dependency project [17]. However, there are more than $6,000$ languages in the world, and we do hope that the computers will be able to analyze all the languages. Even though the resources for a wide range of languages have been developed, the data quantity and quality for many languages are still very low, especially the Southeast Asian languages, e.g., Khmer, Thai, Lao, Burmese, which is hard to achieve a desired performance. In addition, NLP for those languages is not well studied in the past; even their word tokenization is still under development. Certainly, more contributions are required to solve language-specific problems and to enlarge the language diversity of the resources.

The Khmer language, which is the official language of the Kingdom of Cambodia, is studied in this thesis because developments of the morphological and syntactic analyses for this language were rare in the past. The fundamental processing for this language remains challenging and restricted studies of the other

NLP tasks. Previously, there exist several studies on the word tokenization and POS tagging, but most of them are not extendable because the data were mostly closed. For Khmer NLP communities, more data and studies regarding the fundamental processing are required to support the other NLP tasks.

Data development that is based on human annotation can be very costly, depending on the complexity of the tasks. For example, syntactic structure annotation requires linguists to understand the exact meaning of the sentence and then combine or connect words to form a structure to match the sentence's meaning; this process takes a lot of human-annotation efforts and times, especially for long and complex sentences. In practice, pseudo annotations are used to reduce the annotation process such that linguists can simply focus on the checking and editing process. Generally, the pseudo annotations could be produced using either a cross-domain, cross-lingual, or fully unsupervised model. These models could also be used for parameter initialization to obtain high performance with less training data.

This thesis aims to develop the morphological and syntactic analyzers for the Khmer language. Since data are the most important part of the development, we attempt to create and release data to the public together with empirical analyses and discussions on the data for future developments of the analyzers. We also investigate cross-lingual transfer approaches to facilitate data development and model training with less data.

The cross-lingual transfer assumes that there exist projectable semantic units between two languages, and the task to transfer is applicable for the low-resource languages. As a consequence, the cross-lingual transfer is very hard for the morphological analysis because it is nearly impossible to project the morphemes or characters across languages even though the words are translated across language, especially when their scripts do not overlap. As a result, the morphological analysis is regarded as language-specific, where the human annotation is necessary for its development. On the other hand, the cross-lingual transfer is applicable for the syntactic analysis for two reasons. First, meaningful words of one language can generally be translated to the other languages. Second, it is easy to find two languages with the same words order, e.g., *subject-verb-object*. In such case, if two languages are isomorphic, the syntactic structures of their mutually translated

Figure 1.1: Formation of "*minister of defense*" in Khmer. The expression can be decomposed into five morphemes 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 *protect*s the *nation*. The entire expression is also formed head-initially, 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.

sentences are likely similar. However, the cross-lingual transfer assumes that the words can be detected or the morphological analysis exists for each language. For this reason, we discuss the development of the morphological analysis first then explore the cross-lingual transfer for the syntactic analysis after. To conclude, this thesis is organized into two parts, first part presents the data development and discussion towards the morphological analysis for Khmer; the second part explores the cross-lingual transfer for the syntactic analysis for a wide range of low-resource languages in addition to Khmer.

1.1. Morphological Analysis

The Khmer language, as a typical Austroasiatic language, 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 1.1 illustrates basic morphological features of the Khmer language and also 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. As aforementioned that the morphological analysis can be regarded as the cutting-and-tagging task, 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 [18–20]. To accommodate the ambiguity of tokens, Chea et al. [19] 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 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.

This work is one component of the *Asian Language Treebank* (ALT) project [21], and a part of our contribution is a preparation of a systematically annotated 20,000-sentence Khmer corpus with tokenization and POS-tagging information. Compared with a previous corpus by Ye et al. [22]*, which was the only Khmer POS-tagged corpus publicly available, our corpus is superior in terms of the scale and also contains annotation for more linguistic phenomena such as grammaticalization and compounding. Specifically, we used the `nova` annotation scheme [23] that is designed to focus on the highly analytic languages, especially of the Southeast Asian languages. Our manual annotation enjoyed its features, that is, a compact tagset with only four basic and three auxiliary tags, and its concept of functional tags for the grammaticalization phenomena. At the same time, we can annotate POS tags for words and their morphemes within a single corpus using the `nova`'s combination tags—a pair of brackets "`[`" and "`]`", e.g., a word "*going*" is annotated as *go*_`v[v` *ing*_`o-]v` where `v` is a verb and `o-` is a tense marker. For the research community, our corpus has been released under a `CC BY-NC-SA`

*This corpus contains 12,000 POS-tagged full-sentences and people-names.

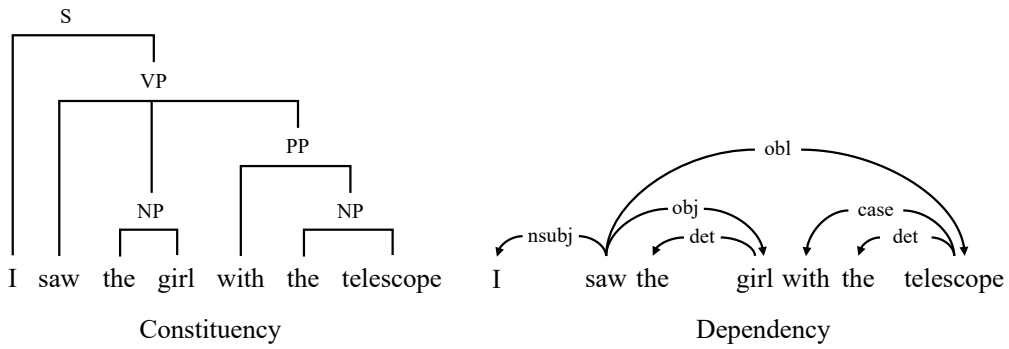


Figure 1.2: Examples of constituency and dependency structures.

license [24].

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 the morphological level. Specifically, the approaches are (1) a point-wise classifier by 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 a 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 [25], this work can be a reference for the development of low-resource languages with similar features widely used in Southeast Asia.

1.2. Syntactic Analysis

There are two widely used syntactic structures, that is, constituency and dependency. A constituency structure or phrase structure describes how words in a sentence are combined while a dependency structure represents the relationship of words. In other words, the dependency structure is one-to-one word correspondence while the constituency structure is hierarchical relations as in Fig. 1.2.

The automatic processing of the syntactic analysis is known as parsing, which is commonly data-driven that a parsing model is learned based on a treebank.

In the multilingual NLP era, the development of multilingual syntactic parsing models is highly demanded. Therefore, the Universal Dependency Project [17] has been introduced and has gathered dependency treebanks for more than a hundred languages. On the other hand, the constituency treebanks are available only for several languages that can be obtained from the Asian Language Treebank project and the Statistical Parsing of Morphologically Rich Languages (SPMRL) share task. Even though recent neural-based NLP systems achieve very high performance, the constituency structures remain beneficial, especially for machine translation [13, 26, 27] for low-resource [11] or zero-resource settings [14]. However, the experiments of these studies are limited to languages whose constituency parsing models are available. For the generalization of the syntax-based NLP systems, the multilingual constituency parsing models are necessary. Because the treebank development is costly, this thesis aims to study the cross-lingual transfer approach for constituency parsing based on currently existing constituency treebanks.

The cross-lingual transfer is a promising approach toward low- or zero-resource problems. Explicitly, a syntactic parsing model for a high-resource (source) language can be used to parse a sentence of another low-resource (target) language through a certain cross-lingual signal. Previous studies mostly investigated the cross-lingual transfer for dependency parsing [28–30] due to dependency-treebanks availability for more than 100 languages under the Universal Dependency Project and those treebanks were annotated following the same guideline. However, there were few studies for the constituency parsing, which could be because the annotation guidelines of the existing constituency-treebanks are different, especially their syntactic label sets that make cross-lingual transfer and evaluation difficult. However, there should exist some universal properties of the constituency structures in the existing treebanks inspired by the Noam Chomsky's universal grammar theory [31].

Therefore, in the latter half of this thesis, we explore the performance of cross-lingual constituency parsing based on the existing constituency treebanks. We investigate two variants of the cross-lingual transfer, which are organized into

two experiments. In our first experiment, we assume the availability of the POS-tagged corpus for each target language and investigate single-source cross-lingual constituency parsing based on this resource condition. Specifically, we propose a cross-lingual constituency parser by delexicalization and investigate the effectiveness of various source-language selection techniques. Even though our results revealed that this setup by delexicalization outperforms fully-unsupervised baselines, this approach remains several limitations i.e., parsing ambiguity of the delexicalized model and limitation of source-language selection. Subsequently, our second experiment addresses the aforementioned limitations and examines multi-source cross-lingual constituency parsing using a lexicalized pretrained multilingual language model. Concretely, we combine constituency corpora in multiple languages and train a single multilingual parser together with a pretrained multilingual language model. Because the corpora are language-specific and the language structures are diverse, we propose a method to preprocess the constituency treebank and to integrate typological features to universalize the constituency parser. Based on experimental results, we revealed that the pretrained multilingual language model is highly effective for cross-lingual constituency parsing together with our preprocessing step and typological integration.

1.3. Contributions

The contributions of this thesis are summarized as follows:

- We develop the largest tokenized-and-POS-tagged Khmer corpus that contains annotation for the linguistic phenomena of the grammaticalization and compounding; present analysis and discussions to show the achievements and limitations of the data and the machine learning approaches.
- We propose the cross-lingual constituency parser by delexicalization and demonstrate the effectiveness of various source-language selection techniques. We show that the model outperforms the fully unsupervised baseline. Additionally, we analyze and discuss the limitation of the delexicalization approach i.e. parsing ambiguity of delexicalized model and limitation of source-language selection.

- We examine the multi-source cross-lingual constituency model using lexicalized pretrained multilingual language models as cross-lingual signals. We illustrate that this model preserves parsing performance for the source language and improves cross-lingual performance even without any bilingual information.
- We propose a treebank preprocessing step and typological features integration to generalize the cross-lingual performance of the multi-source cross-lingual constituency model.

1.4. Thesis Structure

The remaining parts of this thesis are organized as follows: Chapter 2 reviews the Khmer language, the background of the sequence labeling, constituency parsing, and cross-lingual transfer; Chapter 3 describes the development of our two-layer tokenization and POS tagging corpus and discusses its automatic processing for the Khmer language; Chapter 4 describes our cross-lingual constituency parsing by delexicalization; Chapter 5 presents our approach for multi-source cross-lingual constituency parsing; Finally, we conclude the thesis and discuss future works in Chapter 6.

Chapter 2

Background

2.1. The Khmer Language

The Khmer language, which is an official language of Cambodia, is spoken by more than 15 million people in Cambodia (2019 census) [32]. Other minority speakers are in southern Vietnam and northeast Thailand. The Khmer dialects are classified into Central Khmer, Northern Khmer that is spoken in northeast Thailand, Western Khmer, Phnom Penh Khmer, Southern Khmer that is spoken in southern Vietnam, and Khmer Khe. However, the central Khmer is the standard dialect that is taught in Cambodian schools. The Khmer language has been revolutionized and its history is divided into four periods, that is, the Old Khmer period that is subdivided into pre-Angkorian (from AD 600 to 800) and Angkorian (from 9^{th} to 13^{th} century), the Middle Khmer (from 14^{th} to 18^{th} century), and the Modern Khmer (18^{th} century to present). In the Middle Khmer period, the language had a major change in morphology, phonology, and lexicon. In addition, many words were borrowed from Thai and French in this period. In the early 20^{th} century, the language was again changed and was standardized as the modern language under the Khmerization phase, which was a transition phase of getting rid of the foreign elements, reviving affixation, and using the old Khmer roots (from historical Pali and Sanskrit) to develop new words for modern ideas. Until today, the language is recognized as the Modern Khmer.

The Khmer script is based on the abugida writing system, in which consonant-vowel sequences are written as units. The Khmer script consists of 33 consonants, 14 independent vowels, 16 dependent vowels, 13 diacritics [33], and other symbols such as numbers and signs. The consonant is the main letter for each unit while the vowel is the secondary, which is called the dependent vowel. In addition, diacritics are also commonly used in Khmer writing, which are also attached to a

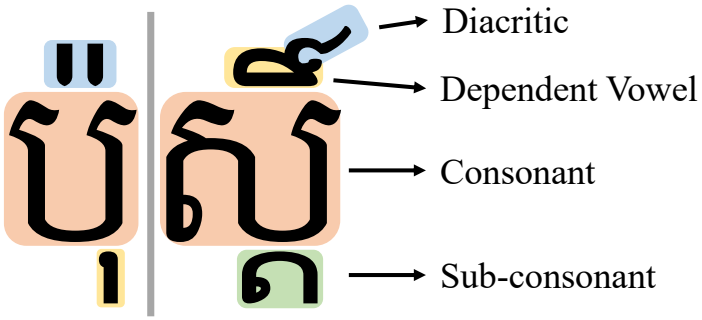


Figure 2.1: A Khmer word example. Its word translation is "post office". The word consists of two consonant-vowel combinations, which are separated by a vertical grey line.

consonant letter similar to the dependent vowel. Furthermore, a consonant letter can be attached with more than one dependent vowel and diacritic. On top of this combination, another consonant can be transformed into a sub-consonant and attached under the main consonant letter, of which example is illustrated in Fig. 2.1 However, unlike the dependent vowels, the independent vowels stand alone without being attached to any consonant letter.

Regarding grammar, Khmer is an analytic language, which has no inflection on words and auxiliary words are used to indicate the tense of sentences. Moreover, Khmer is a zero-copula language where copulas for adjectives (or even a noun) are mostly eliminated. The basic word order follows subject-verb-object and head-initial order. Topic-comment structures and prepositions are also commonly used in this language. Additionally, Khmer morphology was changed, at some points in the past, from an agglutinative to an isolating language, which uses little prefixes or suffixes in the modern Khmer's morphology. However, compounding is very common in the modern Khmer that will be further discussed in Chapter 3.

2.2. Sequence Labeling

Sequence labeling is a type of pattern recognition task that assigns a categorical label to each element of a given input sequence, which is suitable for tokenization and POS tagging tasks. Precisely, the input type and the labels set must be

defined to apply the sequence labeling for the tokenization and POS tagging. For instance, for tokenization, the input and the label set are a characters sequence and the word boundary information (e.g., IBES*), respectively. For POS tagging, the input is a words sequence and the label set is a POS tag set.

Let's denote $\mathbf{x} = \{x_1, \ldots, x_T\}$ and $\mathbf{y} = \{y_1, \ldots, y_T\}$ the input and label sequence, respectively; M the size of the label set such that $y_i \in \mathcal{C} : \{c^1, \ldots, c^M\}$. The lengths of both x and y are assumed to be the same. The label sequence can be determined by estimating a conditional probability distribution $p(y|x)$ and then searching for its optimal sequence as follows

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}) \tag{2.1}$$

In supervised learning, a set of training data $D = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$ is given for modeling the conditional probability distribution $p(y|x)$. This section describes particularly the modeling approaches that are applied in Chapter 3, i.e., support vector machine (SVM), conditional random field (CRF), recurrent neural network (RNN), and neural CRF. In short, for statistical models, SVM is a maximum-margin-based algorithm that requires less computational power where CRF considers the contextual information as well as the labels of its neighborhood to address an ambiguous prediction. For neural models, RNN with LSTM cells can capture wider contexts. A neural CRF model has a CRF layer added on the top of an RNN and predicts each label depending on neighboring predictions.

2.2.1 Support Vector Machine (SVM)

Support vector machine (SVM) is used to classify the data points by determining the optimal boundary—hyperplane—between two classes for binary classification problems. For simplicity, we only discuss the linear SVM, which is used in Chapter 3. For example, given a data point x_i, the corresponding label y_i is determined by a sign of a function $g_{w,b}(x_i) = w^T x_i - b$. The hyperplane lies on a set of points x that satisfies $g_{w,b}(x) = 0$. The hyperplane is $(p-1)$-dimensional where p is the

*The four tags of IBES represent the Beginning of a token, End of a token, Inside a token, and Single unit, which is simultaneously the beginning and the end of a token.

dimension of its data points. In training, the hyperplane is optimized by finding the one that maximizes the distance from itself to its nearest data points. Theoretically, given a set of training data points, $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ and $y \in [-1, 1]$, the hyperplane is detemined by minimizing

$$\min_{w,b} \left(\frac{1}{2}||w||^2 + \lambda \sum_{i=1}^{N} \max(1 - y_i \cdot g_{w,b}(x_i), 0) \right) \tag{2.2}$$

where $\lambda > 0$ is a penalty parameter.

The approach can be later cast to multi-class SVM by breaking down the multiclass problem into multiple binary classifications. There are two common methods for this problem, that is, one-vs-one and one-vs-the-rest. For one-vs-one, $M(M-1)/2$ hyperplanes are needed, and each hyperplane is used for separating two classes. The class of each instance is determined by the votes of this set of hyperplanes. One-vs-the-rest uses one hyperplane for each class that separates itself from the rest of the classes. Therefore, it requires only M hyperplanes. We used the latter approach in this thesis. In one-vs-the-rest, the decision of multi-class classification at position i is by choosing a highest score binary classifier such as

$$\hat{y_i} = \arg\max_{y' \in \mathcal{C}} g^{y'}(x_i) \tag{2.3}$$

The model parameters can be optimized by Crammer-Singer formulation [34] as

$$\min_{w,b} \left(\frac{1}{2}||w||^2 + \lambda \sum_{i=1}^{N} \max(1 - g^{y_i}_{w,b}(x_i) + \max_{y' \neq y_i} g^{y'}(x_i), 0) \right) \tag{2.4}$$

where y_i is a gold label at sequence position i, and y' is any class label in $\mathcal{C}$. Note that each data point x_i needs to be a numerical vector where the actual input is a text. Therefore, we calculate a set of features for each textual input x_i following Neubig and Mori [35]. The features also include the contextual information for each input to hope that classifier could make a better prediction.

2.2.2 Conditional Random Field (CRF)

Conditional random field (CRF) is an undirected graphical model such that a conditional probability distribution $p(\mathbf{y}|\mathbf{x})$ is calculated with respect to the conditional dependence structure of a given graph $G = (V, E)$. A simple linear-chain CRF, which has been applied for many NLP sequence labeling tasks, e.g., POS tagging, shallow parsing, named entity recognition, is described in this section. Precisely, the conditional probability distribution $p(\mathbf{y}|\mathbf{x})$ is written compactly with the concept of feature functions and their corresponding weights. Each feature has a form $f(y_{t-1}, y_t, x_{1:T}, t)$, which can be either a state-observation $s(y_t, x_{1:T}, t)$ or transition feature $t(y_{t-1}, y_t)$ that are equivalent to the vertices V or edges E of the graph G, respectively. Each feature is represented by a Boolean value of either 0 or 1. Therefore, the conditional distribution $p(\mathbf{y}|\mathbf{x})$ is written as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{t=1}^{T}\sum_{i=1}^{K} w_k f_k(y_{t-1}, y_t, x_{1:T}, t)\right) \tag{2.5}$$

where K is the number of all features, $Z(\mathbf{x})$ is a partition function, and w_k is an associated weight of the feature f_k. We can rewrite the equation with a global feature function $F_k(\mathbf{y}, \mathbf{x})$ as follows

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{i=1}^{K} w_k F_k(\mathbf{y}, \mathbf{x})\right) \tag{2.6}$$

$$F_k(\mathbf{y}, \mathbf{x}) = \sum_{t=1}^{T} f_k(y_{t-1}, y_t, x_{1:T}, t) \tag{2.7}$$

The weights $w_k \in W$ are the model parameters and learned from a given data set $D = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$ by maximizing the log-likelihood objective function as

$$\mathcal{L} = \sum_{i=1}^{N} \log p(\mathbf{y}|\mathbf{x}) \tag{2.8}$$

$$= \sum_{i=1}^{N} \left(\sum_{i=1}^{K} w_i F_k(\mathbf{y}, \mathbf{x}) - \log Z(\mathbf{x}) \right) \tag{2.9}$$

An important remark is that the feature function $f(y_{t-1}, y_t, x_{1:T}, t)$ is not necessary to depend on the observation of every time step. Instead, the observation window can be limited to a certain size. For instance, the feature at time t can take into account only its previous x_{t-1} and next observation-state x_{t+1} and can be rewritten as $f(y_{t-1}, y_t, x_t, x_{t-1}, x_{t+1})$. A more sophisticated set of features can be found in many practical NLP applications using CRF [2, 36].

2.2.3 Recurrent Neural Network (RNN)

Both SVM and CRF are based on hand-crafted features. To relax this constraint, a neural network together with deep learning has been applied for the textual sequence labeling task that learns input features instead of the hand-craft feature engineering [37]. Since contextual information is crucial for the prediction, a recurrent neural network (RNN) has been introduced to memorize the internal state of the whole sequence by its dynamic behavior. To control the flow of the memory more efficiently, long short-term memory (LSTM) can be used for the flexibility to either forget or memorize the previous states. In addition, bidirectional RNN encodes a sequence from left-to-right and right-to-left to memorize the context of the longer sequence as in the left of Fig. 2.2. The bidirectional LSTM-RNN model can be formulated as

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp s(\mathbf{y}, \mathbf{x})}{\sum_{\mathbf{y}' \in \mathbf{Y}_\mathbf{X}} \exp s(\mathbf{y}', \mathbf{x})} \tag{2.10}$$

$$s(\mathbf{y}, \mathbf{x}) = \sum_{t=1}^{T} (W([\overrightarrow{h_t}, \overleftarrow{h_t}]) + b) \tag{2.11}$$

where W and b are a weight matrix and a bias vector of an output layer, $\mathbf{Y}_\mathbf{X}$

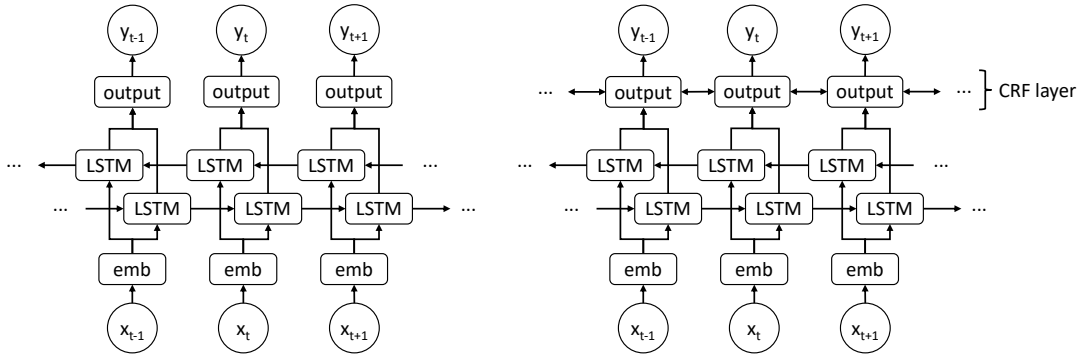


Figure 2.2: General LSTM-RNN architecture.

represent all possible tag sequences for a sentence X, and $\overrightarrow{h_t}$ and $\overrightarrow{h_t}$ are the forward and backward output vectors of the bidirectional LSTM-RNN such as

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(\overrightarrow{h_{t-1}}, embedding(x_t)) \tag{2.12}$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM}(\overleftarrow{h_{t-1}}, embedding(x_t)) \tag{2.13}$$

2.2.4 Neural CRF

The formulation in Eq. (2.11) calculates the score of each label y_t independently from the other output predictions. Instead, it is beneficial to consider the neighbor labels and choose the best chain of the label for a given input sequence. For example, in English POS tagging, it is more likely that an adjective is followed by a noun than a verb. Therefore, a CRF layer can be integrated on top of the RNN model as in Fig. 2.2 and the scoring function $s(\mathbf{y}, \mathbf{x})$ can be rewritten as.

$$s(\mathbf{y}, \mathbf{x}) = \sum_{t=0}^{T} A_{y_t, y_{t+1}} + \sum_{t=1}^{T} (W([\overrightarrow{h_t}, \overleftarrow{h_t}]) + b) \tag{2.14}$$

where A is a matrix of label transition scores. $A_{y_t, y_{t+1}}$ represents the transition score from y_t to y_{t+1} where y_0 and y_{T+1} are the start and end labels of the sequence. Consequently, A is a square matrix of size $T + 2$. The more detail of neural CRF can be found in Lample et al. [38].

2.3. Constituency Parsing

The constituency structure of a sentence describes how words can be recursively grouped as a single unit or constituent that is associated with a syntactic label. The number of constituents is not equal to that of words, which is required a structural model that can generate or parse a constituency structure of a given sentence. There are two mainstream approaches for constituency parsing, that is, transition-based and chart-based parsing. A transition-based model is mainly based on transition states and actions where a set of transition actions decide how a structure could be constructed and tends to produce a well-form output. On the other hand, the chart-based approach, which is also known as dynamic programming, packs all possible structures into a chart table and enjoys searching for an optimal structure.

2.3.1 Transition-Based

The transition-based approach decomposes a sentence's structure into a series of actions such that the task of parsing is to predict those actions for a given sentence [39, 40]. More specifically, a transition-based parser consists of a stack S, a queue W, and a list of transition actions A where S can contain terminal or non-terminal nodes and W contains only terminal nodes. Actions in A can be either *shift* or *reduce* action such that the transition-based parsing is also known as shift-reduce parsing. States of both S and W represent a transition state that will be changed by a transition action in A. At the initial state, S is empty and W contains all the words in order so that the first word is the first element in W. In a final state, parsing is terminated when S has one non-terminal node and W is empty. Furthermore, the states are affected by the transition actions. A shift action takes the first element from W and pushes it into S. Reduce actions are subdivided as unary- and binary-reduce action, which assume the structures are binarized. The unary-reduce action pops each time one element from S, creates a new node on top of it, and then pushes the new node back into S. Like the unary-reduce action, the binary-reduce action takes two elements each time from S, creates a new node on top of them, and then pushes the new node back into S.

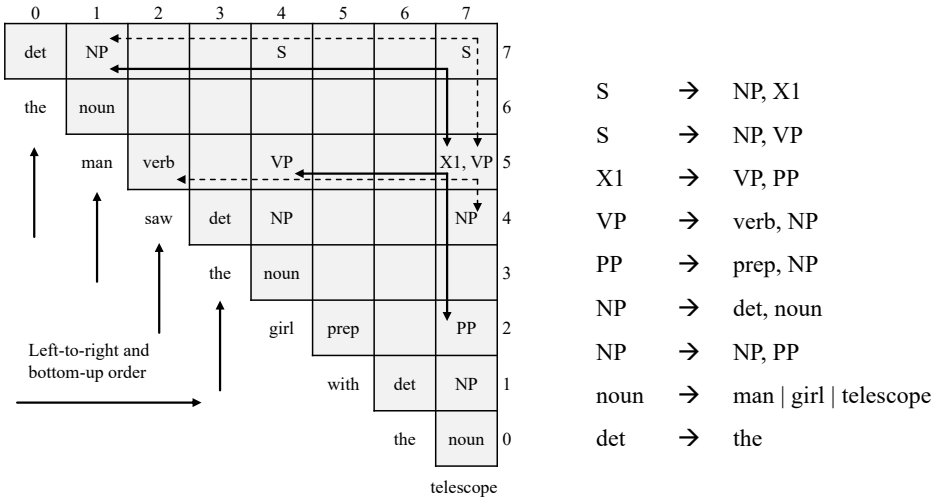
The number of shift actions equals to the number of input words but the number of reduce actions is dynamic that depends on the structure complexity of each sentence.

Statistical Models. A transition-based parser could be a discriminative classifier that predicts each action based on a rich hand-crafted features set extracted from each transition-state [39]. Each parsing output was initially obtained based on a greedy search. However, this may suffer from the error propagation problem that the errors at the early step will affect later predictions. Therefore, beam search can be further applied to prevent such local decision errors. [40, 41].

Neural Models. The transition-based parser was then designed using neural networks. Watanabe et al. [42] extended the parser of Zhu et al. [41] using RNN and attempted to represent the stack and queue using feed-forward neural networks. Cross et al. [43] used a deep LSTM encoder that the stacked-LSTM outputs were used to predict each action instead of feature engineering. There were other strategies that use neural network for the transition-based parser, e.g., recurrent neural network grammar [10] and Tetra-tagging parser [44].

2.3.2 Chart-Based

The Cocke-Kasami-Younger (CKY) algorithm is the most widely used chart-based approach for parsing. The conventional CKY is used for context-free grammars (CFGs), which must be in Chomsky Normal Form (CNF) [45] to make sure that each node only consists of either two non-terminal nodes or a single terminal node. At first, the CKY is used to recognize whether a sentence is valid with respect to a given set of CFGs. For this purpose, a chart table of the sentence is filled based on the given CFGs in a bottom-up fashion. In other words, if we represent the chart table as an upper triangle, the filling processing follows the left-to-right and bottom-up order until the final state is reached. As the example in Fig. 2.3, the triangle table is filled by the left-hand side symbols of the CFG rules until the final state at position $[0, 7]$ is reached, which indicates that the sentence is valid to the grammars. For a parsing purpose, the CKY is extended by creating a node for each table cell and pointing each node to the elements that produce it. Then, parsed structures could be generated by backtracking the pointers of the



a. Chart table b. CFGs

Figure 2.3: An example of the CKY chart table with respect to given context-free grammars (CFGs).

final state node. As the example in Fig. 2.3, the arrows in the chart table are the pointers and two parsed structures are generated for the sentence. One of which is illustrated by solid arrows and the other by dash arrows.

Statistical Models. Probabilistic CFGs (PCFGs) can be further used to produce a score for each node so that the best-parsed structure can be decided by the score of the node at the final state. Extensions of the statistical chart parser focused on refinement of the CFG rules, e.g., head lexicalization [46], and unlexicalized CFGs [47]; or on designing global features [48].

Neural Models. The CKY has recently been integrated with the neural network [49–51], which is known as neural CKY, span-based, or neural chart-based constituency parsing. The key idea is modeling the neural scoring function of each node so that the best-parsed structure can be generated through the backtracking process. The algorithm does not use grammar to constrain how constituent is combined, instead just relying on the neural scorer. As a whole, the model follows the encoder-decoder architecture where the encoder is to project the input into a compact presentation, and the decoder scores each node in the table and searches for the best-parsed structure using the CKY algorithm. The

variants of the model are mainly based on the encoder architecture such as the bidirectional LSTM [49,50] or the self-attention encoder [51]. Noting that a *span*, which is the boundary of a constituent, is equivalent to the *node* in the table and we will use *span* in the rest of this thesis.

Our work is based on the state-of-the-art self-attention-based encoder-decoder architecture of Kitaev et al. [51]. Concretely, the encoder consists of word embedding and self-attention layers to produce the contextual presentation for each word. Word embedding is a table-look-up process that assigns a vector representation for each textual input. The self-attention layer is responsible for the interaction of each input vector with all other vectors, which is based on the attention that the layer generates. At the decoder side, all possible spans are extracted and each span (i, j) is represented by a hidden vector $v_{i,j}$ that is constructed by subtracting the representations associated with the start and end of the span. Then, a labeling score $s(\cdot)$ is assigned to each span (i, j) by an MLP span classifier as

$$s(i, j, \cdot) = W_2 g(f(W_1 v_{i,j} + c_1)) + c_2, \tag{2.15}$$

where W_* and c_* are a weight matrix and a bias vector, respectively; f and g are the layer normalization and ReLU ("Re"ctified "L"inear "U"nit) activation function, respectively. For each sentence, the constituency structure T is represented by a set of labeled spans $\{(i_t, j_t, l_t) : t = 1, \ldots, |T|\}$ where the score of T is

$$s(T) = \sum_{(i,j,l) \in T} s(i, j, l) \tag{2.16}$$

At test time, the optimal structure can be obtained using CKY inference algorithm [52,53]. For training, the model is optimized using a max-margin objective function, the details of which can be found in Kitaev et al. [51].

2.4. Cross-Lingual Transfer

Cross-lingual transfer learning is a task where the available treebanks or models of one language are used to solve the tasks of other languages. *Source* and *target* languages are commonly used in this task where source language refers to a high-resource language that has treebanks or models available, and the target

language is the one with a small treebank or without any treebank. The cross-lingual transfer has been categorized into three groups [54]: annotation projection, treebank translation, and model transfer. In addition, there are more than one source-language treebanks that exist in practice. Therefore, it is necessary to either choose the most relevant treebank to the target language for single-source transfer or combine all of them for multi-source transfer. Thus, this section describes the annotation projection and treebank translation, model transfer, single-source transfer, and multi-source transfer.

2.4.1 Annotation Projection and Treebank Translation

The goals of both annotation projection and treebank translation are similar in that intend to project pseudo structures and use the structures to train a task model for the target language. Concretely, the annotation projection extracts the alignments of words from a given parallel corpus using an unsupervised word alignment algorithm and the partial structures are heuristically projected from source to target based on the alignment results. Certainly, a post-processing step is necessary to normalize the noisy projected structures before training a model for the target language. For treebank translation, a statistical machine translation is trained on the parallel corpus and then used to translate the treebank of the source language. The word alignments between the source sentence and its translation are tracked to perform heuristic projection for the target language. The treebank translation approach was claimed to relax the noisiness of the projected structures. Unlike dependency parsing, very few works were studied for constituency parsing. There were studies for the annotation projection for Chinese and Japanese [55, 56]. After that, there is no further investigation on the annotation projection nor the treebank translation for constituency parsing.

However, the annotation projection and treebank translation strongly rely on the quantity and quality of the parallel corpus and the alignment algorithm. In addition, the parallel corpora for many languages are still low or even unavailable and their partial projected structures are required additional post-processing to train a statistical parser, which is not applicable for the neural-based parser. All

of these constraints make the approaches non-trivial to be reproduced. For that reason, we do not consider the treebank projection and translation in our studies. However, we still believe that the partially projected structures could be useful, e.g., for the model transfer; so, we leave this study for future works.

2.4.2 Model Transfer

The model transfer, which can be regarded as an end-to-end version of the cross-lingual transfer, is a process where a supervised model of the source language is directly applied to solve the task of the target language. The basis of the approach is the use of a cross-lingual representation for the inputs of a parser, such as POS tags for delexicalization [57,58], glossed words [57], or bilingual word embedding that is contextual independent [28] or dependent [59]. Each cross-lingual representation could be produced using different resources. The POS tags obviously need a POS-tagger for each source and target language and their tag sets need to be similar. For glossed words and bilingual word embedding, a bilingual dictionary for a source-target language pair is necessary. The dictionary can be used to translate words from target to source language or reversely for the glossed words and to learn a projection matrix between source and target contextual-independent word embedding [60]. Nevertheless, the projection matrix can be learned in an unsupervised manner [61] or based on Latin numbers or symbols [62] but the performance on the bilingual lexicon induction task is far behind the supervised method using the bilingual dictionary. On the other hand, learning a projection matrix between the contextual-dependent word embedding is more challenging because output embedding for each word is always changed based on its sentence's context, which has been attempted previously [59].

The model transfer has been widely studied for dependency parsing due to the data availability for a wide range of languages and the data annotation follows the same annotation guideline—Universal Dependency Project [17]. For constituency parsing, only Zeman et al. [57] investigated a delexicalized model transfer from Danish to Swedish based on a statistical parser. Therefore, we further investigate the model transfer for constituency parsing for a wider range of languages using the publicly available treebanks as in Chapter 4 and Chapter 5. As this thesis

assumes that a POS-tagged corpus exists for each target language as we have developed one for Khmer, we first investigate the delexicalized model transfer based on the state-of-the-art neural-based constituency parser as in Chapter 4. After that, we further investigate the model transfer using recent pretained multilingual language models that were trained without any bilingual information as in Chapter 5.

2.4.3 Single-Source Transfer

For the single-source transfer, the source-side supervised model is trained using the data of only one source language. As the data for multiple source languages are available in practice, a simple solution is to use English as the source language because the English treebank has been well annotated and studied. However, using English regardless of the target language will suffer from distance language transfer, e.g., between English and Japanese. To avoid this problem, a source language selection technique is necessary and basically a similarity measurement between source and target languages. For the delexicalized model transfer, the similarity measurement can be based on the KullbackLeibler divergence between POS trigram distributions—KLcpos3—of the source and target languages [63]. Another measurement can be based on a language identification tool or a typological database [64]. For a more complicated technique, [65] further trained a source-language ranking model according to the properties of the data and the typological features of the source and target languages. We only consider KLcpos3 and typological-based selection in our single-source transfer scenario in Chapter 4.

2.4.4 Multi-Source Transfer

We can leverage all of the data or models of the multiple source languages by the multi-source transfer approaches that can be subdivided into treebank concatenation [29,30,58,66], model parameters interpolation [67,68], or reparsing [63,64]. The treebank concatenation is to train a single supervised multilingual model using the data of multiple source languages to leverage the structural diversity for the target language. The model parameters interpolation is to combine parame-

ters of the single source parser models were trained using the individual treebanks to obtain a final multi-source model, which is not practicable for neural parsers due to a large number of model parameters. The reparsing is a two-stage approach that multiple single-source parsers are used to generate multiple parses of a target-language sentence that are then combined to weigh each possible substructure so that an optimal structure for the target-language sentence can be extracted. In practice, this approach produces one cross-lingual model for each target language. Compared with treebank concatenation, one cross-lingual model can be used for many languages. Therefore, this thesis further investigates the treebank concatenation for the cross-lingual constituency parsing.

Chapter 3

Tokenization and POS Tagging for Khmer: Data and Discussion

This chapter presents our Khmer tokenized-and-POS-tagged corpus and automatic processing experiments based on this corpus. Firstly, we review related works of the tokenization and POS tagging tasks and previous works on Khmer processing. Secondly, we describe the development of this corpus including the annotation processes and guidelines and illustrate an annotation example that contains the grammaticalization and compounding annotation. After that, based on this corpus, we conduct experiments on four representative machine learning approaches, that is, (1) a point-wise classifier by 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 a 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. Lastly, we present analysis and discussions based on the experimental results to show the achievements and limitations of the data and approaches.

3.1. Related Works

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 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 under-studied and low-resource languages. Unavoidable ambiguities evolved in tokenization and POS-tagging tasks can be further handled by statistical ap-

proaches once annotated data are provided. A classifier trained by SVM may provide a fast solution such as in Neubig et al. [35]. As a more generalized framework for sequence labeling, CRF [69] is considered as a standard baseline. In recent years, neural networks, that is, models that use stacked multiple-layer nonlinear functions [37], have further generalized many typical NLP tasks under a sequence-to-sequence framework based on the methodology of end-to-end processing. The LSTM [70] based RNN has been widely used as a feature extractor to capture long range dependencies within complex sequences. A typical investigation is Yan et al. [71]. Furthermore, the features extracted by an LSTM-based RNN can be concatenated to a CRF interface for a better list-wise search [38, 72–74].

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 Chinese word segmentation [73, 75, 76] and Japanese morphology analysis [1]. 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. [1] for Japanese, a work on joint processing on Chinese is Kruengkrai et al. [77]. Similar to the previous work on Burmese [2], we also discussed the effectiveness of joint processing on a low-resource language. Specifically, two difficulties exist for an under-studied 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. Syntactically, Khmer is more analytic than Burmese [23]. Particularly, grammatical categories for verbal morphemes in Khmer rely substantially on their contexts as further discussed in Section 3.2.4. Compared to the previous work on Burmese, this work investigates the performance of the morphological analysis

regarding this Khmer language features and examines two more approaches, that is, SVM and LSTM-CRF. Moreover, the limitations of both data and approaches are discussed in Section 3.4.

A review of previous work on Khmer processing is as follows. Regarding tokenization, Seng et al. [78] and Bi et al. [18] applied dictionary-based maximum matching; Huor et al. [20] further used corpus-based bi-gram language models; Chea et al. [19] introduced the CRF. Regarding POS-tagging, Nou et al. [79] proposed a template-based approach; Nou et al. [80] 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 contain 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.

3.2. Data

3.2.1 Overview

The Khmer corpus is a component of the ALT project, for which an overview is provided by Riza et al. [21]. Details are also described in previous works on Burmese [2, 81]. 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.* 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. [23] is applied and the annotation guidelines have be released.† In this section, a brief review of the

*Stylistically, 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.

†Annotation 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`

manual annotation process is provided in Section Section 3.2.2, important issues related to the annotation is described in Section 3.2.3, and overall statistics with typical annotation examples are presented in Section 3.2.4.

3.2.2 Timeline

Compared with the previous work on Burmese, where around thirty annotators were involved [2], the team for Khmer annotation comprised only around ten people, and at most times, fewer than five people. Except for Dr. Chenchen Ding, the annotators are Khmer native speakers. All the team members are researchers and students proficient in NLP. It takes around four 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 me and Dr. Chenchen Ding. The annotation was preliminary based on in-house data provided by Chea et al. [19]. Automatic tools for tokenization and POS-tagging were trained and applied to raw Khmer data. One native-speaker annotator (me) began to perform manual checking on the automatically 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 me and Dr. Chenchen Ding with two other native-speaker annotators.

4. In 2019, Dr. Chenchen Ding and I, with another native-speaker annotator, began the syntactic annotation, along which the tokenization and POS-tagging were further polished.

3.2.3 Important Issue

For tokenization, two types of tokens, short 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, 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 [82]; however, it is the minimal unbreakable atom in the writing system that is suitable for textual data processing.[‡]

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 of 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 too complex long tokens. The annotation related to foreign names and loanwords was based on the original languages. Every 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 3.1 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

[‡]Compared with Burmese processing, where the syllable is used as the natural atom [2], Khmer has a more complex syllable structure and less clear boundaries between syllables in writing.

Table 3.1: `nova` tags used in annotation.

	basic tags
`n`	general nouns, can be subjects or objects of tokens tagged by `v`
`v`	general verbs, can take tokens tagged by `n` as arguments
`a`	general adjectives, can directly describe or modify tokens tagged by `n`
`o`	other 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

tags are treated equally. An annotation example is provided in the following Section 3.2.4 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 [2] was applied to improve the consistency among annotators.

3.2.4 Statistics and Example

Table 3.2 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 3.1 shows an example 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

Table 3.2: Statistics for the released ALT Khmer data.

		#token		
dataset	**#writing unit**	**short**	**long**	**#sentence**
`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

case of the `nova` annotation of Burmese [2], the `v-` tag is not used; however, this tag is frequently used for Khmer. This is because Khmer has huge grammaticalized phenomena that many functional particles are derived from basic verbal morphemes [25]. 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 Section 3.2.2 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 Fig. 1.1. 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.

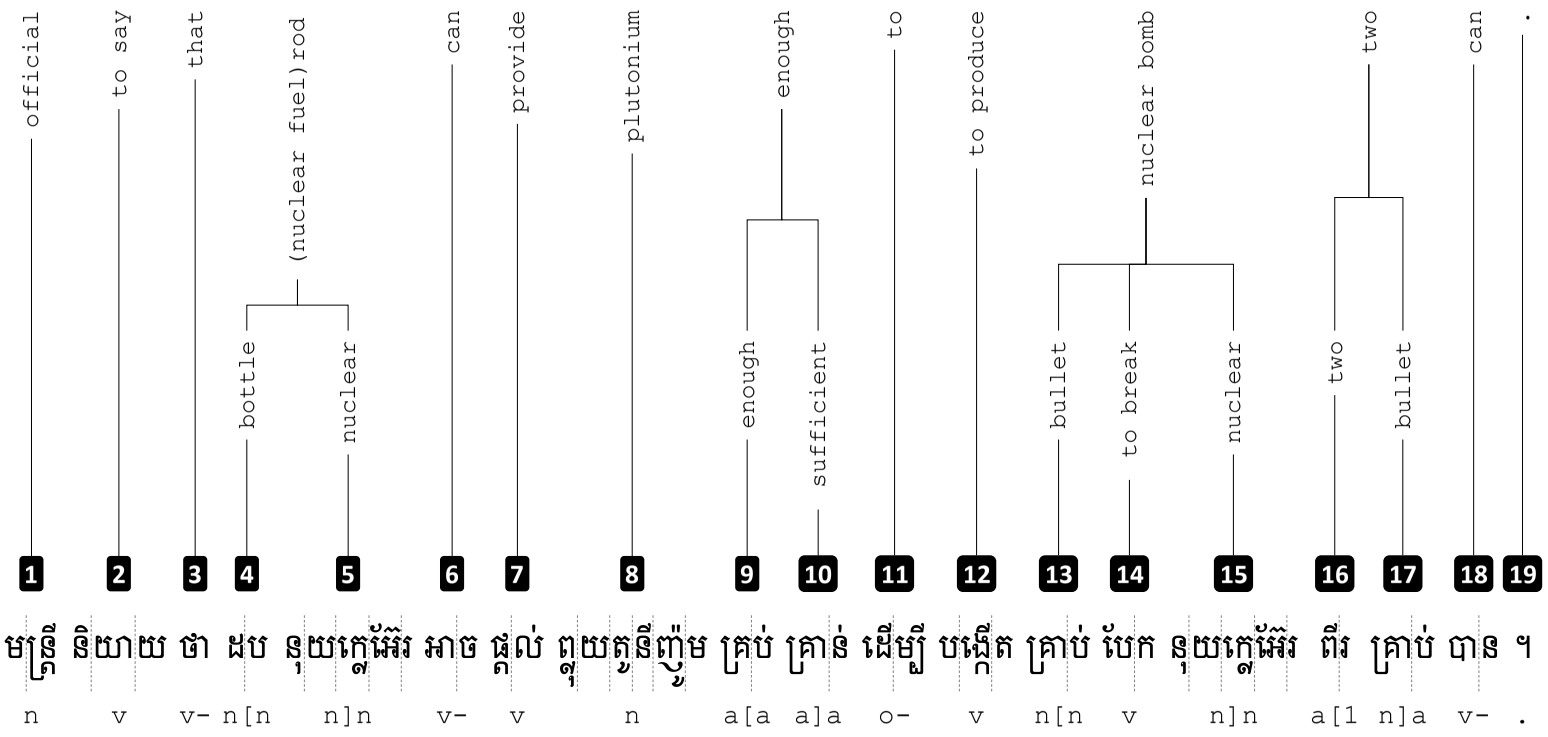


Figure 3.1: Tokenized and POS-tagged Khmer sentence. English glosses for Khmer short tokens and bracketed long tokens are attached at the top of the figure. The boundaries between writing units within each short token are illustrated by vertical broken lines. The original English sentence is "*Officials say the rods could be able to provide enough plutonium to make two nuclear bombs.*"

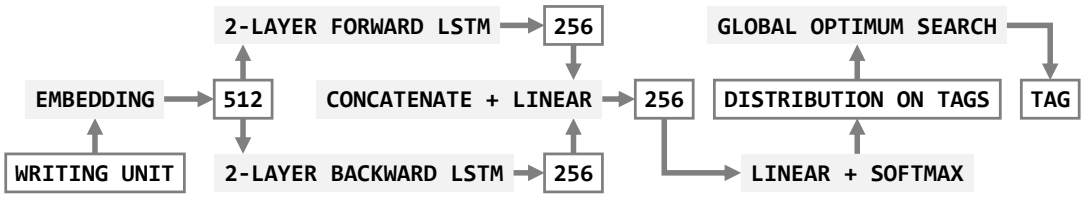


Figure 3.2: Configuration of the network of LSTM-based RNN used in the experiment. The numbers represent the vector dimensions.

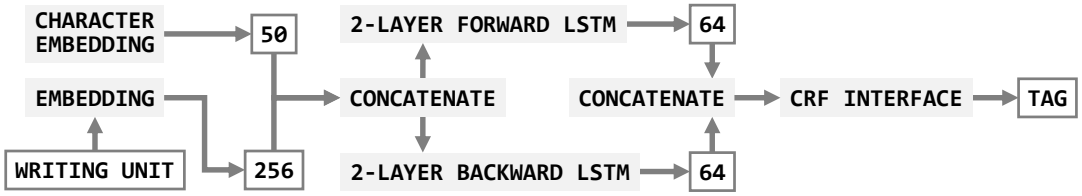


Figure 3.3: Configuration of the network of LSTM-CRF used in the experiment. The numbers represent the vector dimensions.

3.3. Experiment

3.3.1 Settings

As mentioned in Section 3.2.4, 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 **i**nside a token, **b**eginning of a token, **e**nd of a token, and **s**ingle unit, was used by default. Simplified schemes of IE and IB were also compared in experiments. As to the configuration of processing, cascaded as well as jointed manners were compared. The former one is to tokenize first and then to tag the tokens using two separate models; the latter 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 Kytea[§] [83] was used as a preliminary fast solu-

[§]http://www.phontron.com/kytea

tion. The SVM performs point-wise predication for each writing unit with a sliding window of contextual features. Bi-, tri-, and 4-gram features were experimented. As a trade-off of the speed, 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++ toolkit[¶] was set as a standard sequence labeling baseline. The settings and feature template followed the previous work [2].

- An LSTM-based RNN was implemented using DyNet[‖] [84]. The network was essentially configured according to Ding et al. [2], whereas the dimensions of layers were enlarged to achieve better performance. A diagram of the network configuration is shown in Fig. 3.2. The parameters were trained using Adam [85] with the default setting in DyNet.[**] The model ensemble was also conducted, up to 100 models, as large-scale model-ensemble was found gradually increase the performance.

- An LSTM-CRF was implemented using NCRF++[††] [86]. The network configuration followed that in Ma et al. [73] with the dimension of layers adjusted to fit our task. A diagram of the network configuration is shown in Fig. 3.3. The stochastic gradient descent method is used to optimize the model with the default setting in NCRF++.[‡‡] 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

[¶] `https://taku910.github.io/crfpp/`

[‖] `https://github.com/clab/dynet`

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

[††] `https://github.com/jiesutd/NCRFpp`

[‡‡] 10 for batch size and 0 for momentum. Initial learning rate $\eta_0 = 0.015$ and decay rate $\rho = 0.05$.

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.

3.3.2 SVM

Table 3.3 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 that 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.

3.3.3 CRF

Table 3.4 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.[§§]

[§§] As an interesting side finding, the `IE` scheme was slightly better than the `IB` scheme, particularly for long tokens. In Ding et al. [2], 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 3.3: F-score 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	.971	.870	.951	.851	.972	.883	.946	.851
3-GRAM/IBES	.977	.880	.953	.858	.976	.887	.951	.860
4-GRAM/IBES	.976	.882	.952	.855	.976	.888	.951	.862

Table 3.4: 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	.980	.875	.959	.855	.980	.886	.959	.873
3-GRAM/IBES	.982	.896	.961	.876	.980	.896	.962	.883
4-GRAM/IBES	.982	.899	.960	.878	.979	.898	.960	.884
2-GRAM/IB	.969	.837	.948	.818	.972	.870	.951	.856
3-GRAM/IB	.977	.885	.956	.865	.976	.886	.958	.872
4-GRAM/IB	.977	.888	.956	.868	.975	.888	.957	.874
2-GRAM/IE	.971	.851	.950	.831	.974	.879	.953	.865
3-GRAM/IE	.976	.889	.955	.869	.976	.892	.957	.878
4-GRAM/IE	.976	.890	.956	.870	.975	.891	.957	.877

Table 3.2 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 .961 to .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

.878 to .884 in terms of POS-tagging, but no difference for tokenization (.899 and .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 3.4 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 .81. Therefore, even though the gain that resulted from joint processing was limited, it was indispensable for long token processing.

3.3.4 RNN

Table 3.5 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 .983 for tokenization and .967 for POS-tagging. The improvement for tokenization was limited compared with the CRF's best results of .982; the improvement for POS-tagging was slightly larger: from the CRF's .962 to .967. For long tokens, the best performance for tokenization was .906, and for POS-tagging was .892. Comparing the respective best results of .899 and .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.

3.3.5 LSTM-CRF

The results by LSTM-CRF are presented in Table 3.6. 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., .983 vs. .983 on tokenization and .967 vs. .967 on POS-tagging. On long tokens, the ensemble improved the best RNN model from .892 to .906 (+.008), while only from .898 to .900 (+.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 .871 to .892 (+.021) while only .885 to .888 (+.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 [87] 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.

3.4. Discussion

3.4.1 Short Token

As the RNN and LSTM-CRF had a 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 Fig. 3.4. 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; the duplicated expression of the preposition

Table 3.5: 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	.975	.883	.955	.866	.977	.879	.954	.856
MAXIMUM @100	.979	.891	.961	.878	.979	.892	.960	.871
10-ENSEMBLE	.981	.903	.965	.890	.983	.903	.966	.888
20-ENSEMBLE	.982	.902	.966	.890	.983	.906	.967	.891
50-ENSEMBLE	.982	.904	.966	.891	.983	.906	.967	.891
100-ENSEMBLE	.982	.904	.966	.891	.983	.906	.967	.892

Table 3.6: 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	.978	.887	.962	.875	.979	.892	.961	.878
MAXIMUM @100	.981	.894	.965	.881	.982	.898	.964	.885
10-ENSEMBLE	.982	.895	.965	.884	.982	.898	.966	.885
20-ENSEMBLE	.982	.897	.966	.885	.982	.899	.966	.886
50-ENSEMBLE	.982	.898	.967	.886	.983	.900	.967	.888
100-ENSEMBLE	.982	.897	.966	.885	.983	.900	.966	.888

"*in*". Strictly, these types of over-segmented[¶¶] 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*"

[¶¶]If 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".

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.

In joint processing, the errors were mostly related to POS-tagging rather than tokenization. Typical words with confusing POS tags are listed in Fig. 3.5, 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*",[***] and "*to go to*",[†††] 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.

3.4.2 Long Token

As long tokens cover a large range of complex phenomena, a lexicalized statistic results in a long tail of singletons. Table 3.7 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.

[***] A copula and somehow a grammaticalized instrumental case marker.
[†††] Somehow, a grammaticalized dative case marker.

manually segmented token		automatically segmented token			
រថភ្លើង	train	រថ	vehicle	ភ្លើង	fire
ម្ចាស់ក្សត្រី	queen	ម្ចាស់	owner	ក្សត្រី	queen
នៅក្នុង	in	នៅ	at	ក្នុង	in
ត្រែនស្ទីស្ត្រីន	Transnistria	ត្រែនស្ទី	*transni-	ស្ត្រីន	*-stria
ជេរូសាលឹម	Jerusalem	ជេរូ	*jeru-	សាលឹម	*-salem

Figure 3.4: 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

Figure 3.5: Distribution of manually annotated POS tags for short tokens difficult to process automatically.

Table 3.7: Error distribution for long tokens at the POS-tagging level and for the number of tokens. 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.

	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%

manually segmented token				automatically segmented token			
ប្រាក់	money	ឈ្នួល	salary	ប្រាក់ឈ្នួល	salary		
ប៉ូល	pole	ខាងជើង	north	ប៉ូលខាងជើង	the North Pole		
ក្រុមអូស្ត្រាលី	Australian team			ក្រុម	team	អូស្ត្រាលី	Australia
ការប្រកួតស៊ើរី	series match			ការប្រកួត	match	ស៊ើរី	series

Figure 3.6: 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 Fig. 3.6. The example of "*salary*" is similar to the example of "*queen*" in Fig. 3.4; 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.

3.4.3 Data Size and Further Improvement

Figure 3.7 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

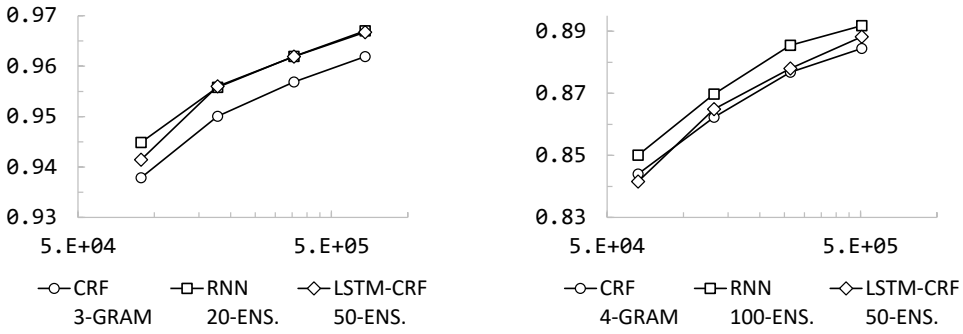


Figure 3.7: F-score for joint processing on short (left) and long (right) tokens on different training data sizes. (x-axes for the number of tokens, in logarithmic scale.)

data set.

It 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 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.

3.5. Summary

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 paper 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.

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 under-studied in the NLP field.

Lastly, we have extended this corpus for the following chapters. For instance, we have developed a small amount of treebank on top of this corpus to facilitate the cross-lingual experiments of the following chapters. In addition, we have converted this POS-tagged corpus from the `nova` tags to the universal POS-tags for Chapter 4 based on the naive projection by Ding et al. [23] and with additional manual conversion mostly on the distinction between the prepositions and functional-verbs.

Chapter 4

Cross-Lingual Constituency Parsing by Delexicalization

This chapter presents cross-lingual constituency parsing by POS-based delexicalization. We assume the availability of the POS-tagged corpus for each target language as we have already developed one for Khmer. Furthermore, single-source transfer scenario, which considers only one source language each time, is considered and various source-language selection techniques are investigated in this chapter. Nevertheless, the aim here is to observe the cross-lingual performance of delexicalized models throughout analyses and discussions on how to maintain the best performance and what limitations the models have. Additionally, because reproducing cross-lingual transfer baselines, e.g., treebank projection [55, 56], is non-trivial as mentioned in Section 2.4 and we do not assume the availability of parallel corpora for each source-target languages pair, we compare the delexicalized models with the fully-unsupervised models, of which intention is to show which models should be used to produce pseudo annotation for the future constituency-treebank development.

4.1. Related Works

Grammar induction, a fully unsupervised constituency parsing, has a long history of attempting to induce constituency structure on sentences in a plain corpus. The early successful attempts were the statistical approaches introduced by Clark et al. [88] and Klein et al. [89]. In particular, Clark et al. [88] tried to induce PCFG by clustering POS tag sequences according to mutual information, that is, their contexts. Based on the same motivation—that tag sequences with the same context are likely to be constituents—Klein et al. [89] introduced a generative constituent-context model that can be optimized by an EM algorithm. Recently,

neural networks have been used for grammar induction. Shen et al. [90] induced constituency structure by simultaneously modeling a language and predicting the syntactic distance between words. Drozdov et al. [91] integrated a recursive autoencoder with an inside-outside algorithm. Kim et al. [92] trained a recurrent neural network grammar with a structured inference network and amortized variance inference. Kim et al. [93] parameterized PCFG with a neural network where the rule probabilities are based on distributed representations, NPCFG, and can be modulated by per-sentence continuous latent variable, CPCFG. NPCFG and CPCFG have become the state-of-the-art models in grammar induction research. Our study compared the cross-lingual model with the state-of-the-art NPCFG and CPCFG models.

Cross-lingual transfer approaches, in which the data or model of one language are used for a model of another language, have been categorized into three groups: annotation projection, treebank translation, and model transfer [54]. Previous studies projected annotation across languages using a word-alignment algorithm for annotation projection [55,56,94] or statistical machine translation for treebank translation [95]. In contrast, model transfer directly takes a supervised model of the source language to apply to plain text in the target language. The basis of model transfer is the use of a shared input representation, such as POS tags [57,58], glossed words [57], or bilingual word embedding [28]. Extensions of model transfer include source-language selection [63, 64], multi-source combination [58], parameter sharing [96], and self-training [97] and others. Our work is similar to that of Zeman et al. [57], which also uses a delexicalized model for transfer learning. However, we explore more languages than the work of Zeman et al. [57], who only considered a single closely-related language pair. Essentially, languages selected for this work have structural diversity. In addition, we select the best source language with or without using the treebank of each target language.

For source-language selection, Rosa et al. [63] introduced a corpus-based distance metric, KLcpos3, which is the KullbackLeibler divergence between the POS trigram distributions of the source and target languages. Agi et al. [64] proposed to select the source language using a language identification tool, langid.py, and a typological database, WALS. Lin et al. [65] trained a source-language ranking model according to the properties of the data and the typological features of the

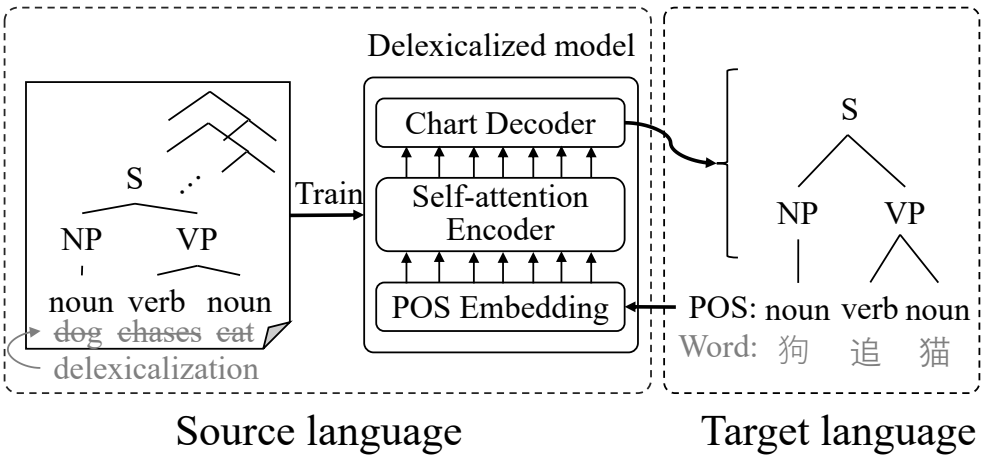


Figure 4.1: Overview of our delexicalized model for cross-lingual constituency parsing. The model is trained on delexicalized structural data of source language and then applied on part-of-speech sequences of the target language.

source and target languages. In this work, we select the best source language simply by using KLcpos3 and the typological features of languages. We also propose a scenario in which a small constituency dataset of the target language, which can be quickly created within a day, is used for source language selection.

4.2. Proposed Methods

In this work, we focus on transferring a constituency parser between languages and selecting the best source language when multiple source languages are available. In the following section, we describe the design of our delexicalized model and the source selection techniques used in this study.

4.2.1 Delexicalized Parser

Our delexicalized parsing model (DEX) is based on the self-attentive span-based parser [51]. Specifically, as shown in Figure 4.1, a self-attention encoder is used to project the input representation and the encoded outputs are combined to represent span. Each span is independently labeled and scored, and then a tree is

incrementally constructed using the CKY algorithm; this is also known as chart parsing. Only POS embeddings are used as input representations, so that the treebank of the source language or the sentences of the target language have to be delexicalized during both training and decoding.

There are two reasons for this span-based parser being suitable for a cross-lingual model. First, a self-attention encoder can capture global context information well and is less sensitive to word order [28]. Second, each span is independently labeled, without considering the label decision of its children or parent [50]. This means that the failure of label prediction on a certain span does not affect label prediction on the other spans. Intuitively, the prediction error caused by local syntax variation between two languages does not strongly affect the overall prediction.

For delexicalization, we use the universal POS tag set introduced by Petrov et al. [98]. The tag set consists of 12 categories: NOUN (nouns), VERB (verbs), ADJ (adjectives), ADV (adverbs), PRON (pronouns), DET (determiners), ADP (prepositions or postpositions), NUM (numerals), CONJ (conjunctions), PRT (particles), PUNC (punctuation marks), and X (a catch-all tag). We used these coarse tags because any language-specific tags can easily be mapped to them. In addition, they cover the most frequent POS tags that exist in most languages [98].

We use the same hyperparameters as Kitaev et al. [51]. It is worth noting that the encoder and embedding layers must have the same dimension. Therefore, the embedding dimension for POS tags was enlarged to have the same size as the encoder. Specifically, we set the encoder to have dimension 1024, with eight layers, eight attention heads, and dimension 64 for key, query, and value. The feedforward layer size is 2048 and the batch size is 250.

4.2.2 Source Language Selection

Constituency data are available for multiple languages, and selecting the best source language is most often better than concatenating the data of all languages to train a single model [64]. We compare two scenarios: first, without awareness of the target language structure and, second, when a small constituency dataset of the target language can be created.

For the first scenario, we compare two language distance metrics: KLcpos3 [63] and the typological feature-based metric. We directly use the precomputed syntactic distance of Littell et al. [99] as the typological feature-based metric. The precomputed distance has only two decimal places and many pairs of languages have the same value. In this case, we use KLcpos3 to further weigh the same distance values to obtain the final best source language. For notation, we use KL for KLcpos3 and SD for precomputed syntactic distance [99].

For the second scenario, we propose to sample 10 sentences whose length is between 10 and 20 and then manually annotate them (L20). We argue that these data can be easily constructed within a day. Subsequently, a language is selected as the best source language if its model achieves the highest unlabeled F1 score on this dataset.

4.3. Experiments

We first parsed each target language using parsers of the other six languages. We then selected the best parser to compare with the following baselines.

4.3.1 Settings

Dataset: The evaluation was performed on seven languages: English (`en`) from Penn Treebank (PTB) [100]; Chinese (`zh`) from Chinese Penn Treebank 5.1 (CTB) [101]; French (`fr`) [102] and German (`de`) from the SPMRL 2013 shared task* [103]; and Japanese (`ja`), Khmer (`km`), and Myanmar (`my`) from Asian Language Treebank (ALT) [21]. Standard splits of each treebank were applied to prepare the training, validation, and test datasets. Table 4.1 presents some statistics of the datasets, as well as their sentence-level word order.

We mapped POS tags of English, Chinese, French, and German using a mapping table provided by Petrov et al. [98]. We created our own mapping table for Japanese. Such mappings were not required for the Khmer and Myanmar datasets because their POS tags already follow the universal tag set. In ALT, Khmer and

*We only considered two of the seven languages in SPMRL because the POS tag mapping of the other five languages is difficult for us.

Order	Language	Train	Valid	Test
SVO	English	39,832	1,700	2,416
	Chinese	18,096	352	348
	French	14,759	1,235	2,541
	Khmer	8,832	512	658
SOV	Japanese	17,202	953	931
	Myanmar	18,088	1,000	1,018
Mix	German	39,465	4,730	4,882

Table 4.1: Data statistics. The numbers refer to numbers of sentences and 'Mix' indicates that word order is a mixture of V2 and SOV.

Myanmar data are analyzed down to the morpheme level [23]. However, we only considered word-level tokens because many morphemes in these languages do not correspond to other languages, which makes cross-lingual transfer very difficult. The POS tags of both source and target languages are manually tagged for this experiment.

Baselines: Two state-of-the-art grammar induction models, NPCFG and CPCFG [93], were used for comparison. Model configuration and implementation[†] mostly followed Zhao et al. [104], who has further evaluated the model for a wider range of languages. We reproduced the results of Zhao et al. [104] and found that the results could be improved for the SPMRL dataset by removing root nodes in the evaluation. We also conducted the experiment with three new languages: Khmer, Japanese, and Myanmar. We additionally trained delexicalized versions of both NPCFG and CPCFG models on the same POS sequences as our model. Following the preprocessing step in the original work, punctuation and trivial spans (width-one and sentence-level spans) were removed before training the NPCFG and CPCFG models. We also compared naive systems, namely, left-branching (LB) and right-branching (RB).

Evaluation: For comparison, the outputs of DEX are post-processed including removal of trivial spans (width-one/unary and sentence-level spans) and punctuation. Additionally, sentences with a length of 1 were excluded from the eval-

[†]`https://github.com/zhaoyanpeng/cpcfg`

		Target Languages						
		en	zh	fr	km	de	ja	my
Source Languages	en	**85.7**	48.6	49.9	47.6	42.3	24.0	24.2
	zh	60.7	**76.5**	34.0	36.5	37.9	21.5	23.5
	fr	45.9	26.2	**71.4**	35.3	38.0	15.7	12.9
	km	42.8	39.1	26.6	**70.0**	29.1	24.6	26.0
	de	40.3	35.3	33.8	27.2	**82.3**	20.4	17.3
	ja	33.0	30.6	25.7	17.3	27.0	**71.3**	50.6
	my	31.8	28.7	27.2	21.9	27.4	53.6	**79.0**
	min	31.8	26.2	25.7	17.3	27.0	15.7	12.9
	max	60.7	48.6	49.9	47.6	42.3	53.6	50.6

Table 4.2: Unlabeled F1 results for all source-target language pairs. Each row represents the source language for training the delexicalized model and each column represents the target language on which the delexicalized model performed prediction. Each score is the average of four runs with different random seeds. Boldface numbers are shown when the source and target languages are the same, and underlined numbers are the best scores for each target language.

uation. Finally, each model is evaluated on sentence-level unlabeled F1 against non-binarized gold trees following the original work of Kim et al. [93].

4.3.2 Results

Table 4.2 shows the results for all source-target language pairs; each result is the average of four runs with different random seeds. The standard derivation over all language pairs is only 0.37, on average, and the minimum and maximum standard deviations are 0.02 and 2.41, respectively. The highest standard deviation is for the Khmer-to-Chinese pair, whereas the others are less than 1.

The diagonal scores can be regarded as the performance of the supervised delexicalized model (DEX Sup) of each language. The performance of our parsing model has been reduced by delexicalization. Compared with labeled F1 results for the lexicalized model on English (93.6), Chinese (87.4), French (84.1), and

German (87.7) without external pretrained embedding[‡], the performance was reduced by between 5.4 and 12.7. The reduction could be larger if our models were measured with span labels because the labeled F1 took into account the label errors.

For cross-lingual performance, the DEX model of English yielded the best performance for Chinese, French, Khmer, and German, but not for Japanese or Myanmar. However, Japanese and Myanmar were mutually the best source languages. The best results for each target language are marked by underlined unlabeled F1 numbers. The sentence-level word order of Japanese and Myanmar are Subject-Object-Verb (SOV), whereas four other languages are Subject-Verb-Object (SVO) and German is a mixed-order language, between verb-second (V2) and SOV order. The results show that the performance of the DEX model fundamentally relies on the word orders of the source and target languages. For instance, at rows six to seven in the first five columns, when Myanmar and Japanese were used as the source languages for SVO languages, the unlabeled F1 values were the lowest or the second-lowest. This indicates that cross-lingual transfer is more difficult when the sentence-level word orders of the source and target languages differ.

The last two rows show the minimum (min) and maximum (max) scores over all choices of source languages, excluding the target language itself. The differences between min and max are very large, averaging 28.1, with a min-max range of 15.4–37.9. This suggests that the ability to select the best source language is highly beneficial for cross-lingual transfer. For each target language, we can determine the best source language based on the max results, denoted as DEX@1.

Data efficiency. Even though language characteristics can be used as a clue to choose the appropriate source language, the quantity and quality of the training data of source languages is also crucial for cross-lingual transfer. We tested this hypothesis by reducing the amount of source-language training data and limiting the sentence length. In particular, we sampled $5,000$ sentences of the training data of each source language, with and without a constraint on the sentence length to select only sentences with a length between 10 and 30. This constraint was

[‡]These results are taken from Kitaev et al. [51] except for Chinese, which is from rerunning the code.

trg	DEX Sup			DEX@1		
	Full	Δ5k	Δ30	Full	Δ5k	Δ30
en	85.7	5.4	25.6	60.7	0.9	16.2
zh	76.5	5.8	25.1	48.6	1.5	13.4
fr	71.4	13.8	24.2	49.9	3.5	17.7
km	70.0	2.3	16.3	47.6	−0.5	9.3
de	82.3	28.3	26.7	42.3	1.7	10.3
ja	71.3	3.1	23.3	53.6	0.2	14.3
my	79.0	2.3	20.4	50.6	1.3	12.9
avg	–	8.7	23.1	–	1.2	13.5

Table 4.3: Unlabeled F1 differences when the dataset sizes were reduced and sentence lengths limited. Δ refers to the reduction in unlabeled F1 score from the model trained on the full dataset to that trained on constrained data.

also applied to the validation set with a sample size of 150. Table 4.3 presents the performance reduction of DEX Sup and DEX@1 when each model was trained on only $5,000$ samples without length constraints (Δ5k) and with length constraints (Δ30). The results show that the performance of both DEX Sup and DEX@1 was worse when the amount of training data was reduced, and the reduction was more obvious when the training data contained only sentences with a very limited length, from 10 to 30. This indicates that the quality and quantity of training data are important for model transfer, as well as for the supervised model itself.

Source selection. Table 4.4 shows the effectiveness of the source-language selection algorithms, KL, SD, and L20. Based on this result, KL and SD were able to correctly pick the best source language for five out of seven languages, and the selection by L20 was identical to DEX@1 as expected. We find that both KL and SD chose German for English and French and their second choice was English. We hypothesize that KL and SD can be improved by filtering source languages from the pool according to their sentence-level word order; for instance, the source-language pool for SVO languages should contain only SVO languages. Figure 4.2 further shows that the closeness of the POS sequences between source

trg	DEX@1		ΔKL	ΔSD	ΔL20
en	60.7	zh	−20.4	−20.4	0.0
zh	48.6	en	0.0	0.0	0.0
fr	49.9	en	−16.1	−16.1	0.0
km	47.6	en	0.0	0.0	0.0
de	42.3	en	0.0	0.0	0.0
ja	53.6	my	0.0	0.0	0.0
my	50.6	ja	0.0	0.0	0.0

Table 4.4: Unlabeled F1 results of source language selection algorithms, where the relative F1 of ΔX = DEX@1 − X.

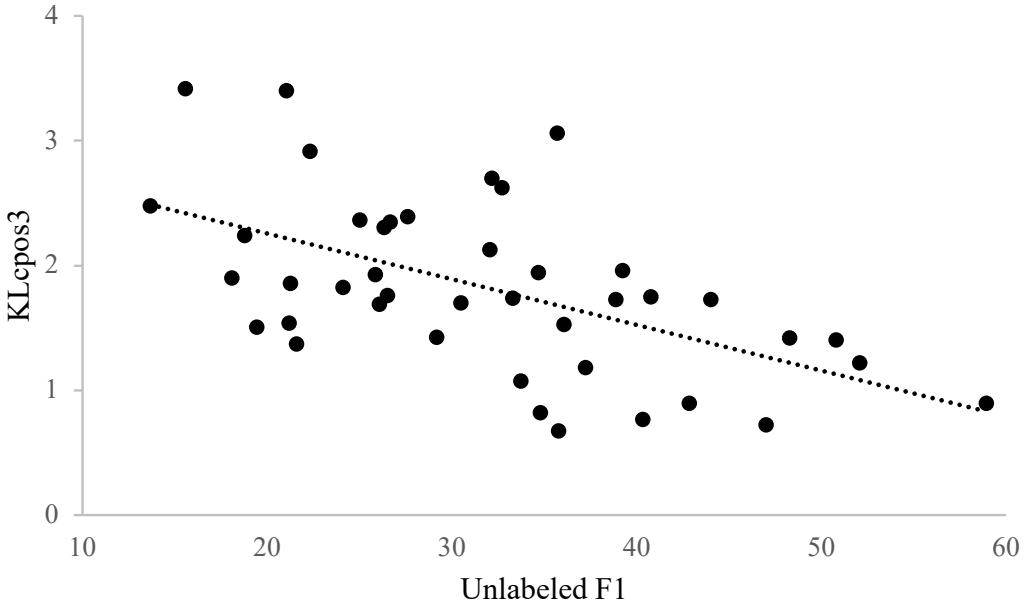


Figure 4.2: Relation between KL and unlabeled F1. Each dot represents a relation for a source-target language pair where source ≠ target.

and target languages might not have an impact on the parsing performance as the Pearson correlation coefficient between KL and unlabeled F1 is only −0.56.

Comparison. We assume that, for each language, we can select the best source language, which we call DEX@1. We also add the second-best language

Model	en	zh	fr	km	de	ja	my
LB	8.9	7.2	7.2	6.9	13.0	14.3	17.6
RB	39.7	25.5	29.1	43.9	18.6	12.8	12.8
NPCFG†	$35.2_{\pm 6.8}$	$26.7_{\pm 1.3}$	$41.0_{\pm 2.5}$	$33.6_{\pm 4.0}$	$29.1_{\pm 11.3}$	$18.1_{\pm 3.8}$	$30.4_{\pm 7.3}$
CPCFG†	$40.2_{\pm 2.0}$	$32.8_{\pm 1.2}$	$42.7_{\pm 1.9}$	$28.2_{\pm 0.7}$	$42.4_{\pm 1.3}$	$19.8_{\pm 3.6}$	$36.6_{\pm 0.9}$
NPCFG	$50.0_{\pm 3.3}$	$29.3_{\pm 4.3}$	$45.7_{\pm 0.8}$	$21.0_{\pm 4.3}$	$41.2_{\pm 0.6}$	$31.8_{\pm 9.3}$	$13.6_{\pm 1.0}$
CPCFG	$56.2_{\pm 1.7}$	$30.7_{\pm 7.4}$	$43.5_{\pm 0.3}$	$24.1_{\pm 4.4}$	$\mathbf{43.5}_{\pm 1.0}$	$24.4_{\pm 7.0}$	$45.8_{\pm 2.4}$
DEX@1	$\mathbf{60.7}_{\pm 0.2}$	$\mathbf{48.6}_{\pm 0.5}$	$\mathbf{49.9}_{\pm 0.3}$	$\mathbf{47.6}_{\pm 0.4}$	$42.3_{\pm 0.5}$	$\mathbf{53.6}_{\pm 0.1}$	$\mathbf{50.6}_{\pm 0.3}$
DEX@2	$45.9_{\pm 0.4}$	$39.1_{\pm 2.4}$	$34.0_{\pm 0.5}$	$36.5_{\pm 0.4}$	$38.0_{\pm 0.4}$	$24.6_{\pm 0.2}$	$26.0_{\pm 0.5}$

Table 4.5: Unlabeled F1 results of all baselines compared with the best DEX model. LB and RB refer to the left- and right-branching baselines. Unlabeled F1 results of NPCFG and CPCFG are the average scores from four runs with different random seeds, and the numbers preceded by $\pm *$ are their standard deviations. † marks the delexicalized version of either the NPCFG and CPCFG model.

(DEX@2) for comparison. Table 4.5 presents the performance of all baselines and the comparison with our DEX models. First, for LB and RB, it is obvious that the languages in the first four columns are strongly biased toward right branches, whereas the other three languages are relatively equally balanced between the left and right branches. From this observation, we can predict that the constituency structures of SVO languages are right-branch-biased and those of non-SVO languages are more equally balanced between the left and right branches. This also supports our argument that sentence-level word order indicates the structural similarity between languages. For the NPCFG and CPCFG baselines, the delexicalized versions of the NPCFG and CPCFG have lower performance than their lexicalized versions in most cases. Therefore, we will consider only their lexicalized versions for the following comparison. To this end, with the exception of German, our DEX@1 outperformed all the baselines by a large margin. Specifically, compared with NPCFG and CPCFG, the improvement in unlabeled F1 score ranges from 4.2 to 37. For German, the unlabeled F1 score of DEX@1 is slightly lower than that of CPCFG but higher than that of NPCFG. Nonetheless, NPCFG and CPCFG were very competitive in comparison with our DEX@2.

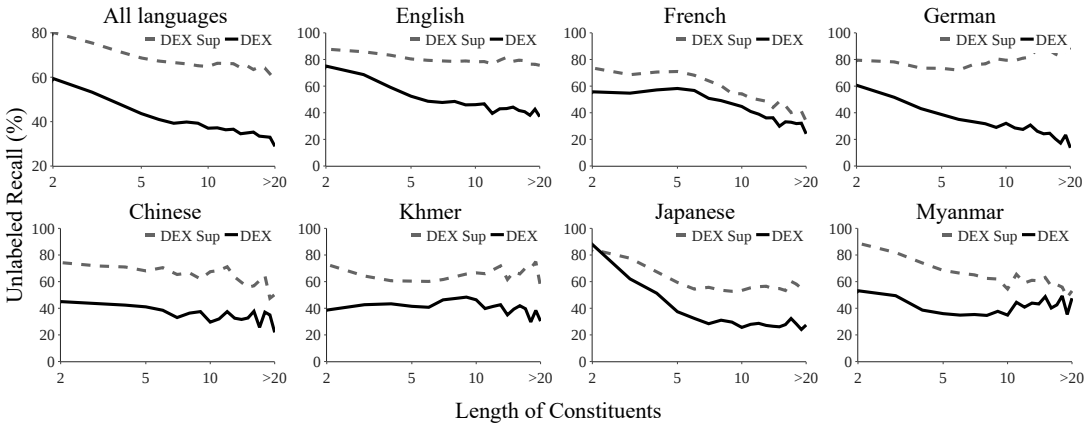


Figure 4.3: The unlabeled recall of the target languages with respect to their constituent length. >20 is the average recall of all constituents with a length greater than or equal to 20.

There was a considerable reduction in performance from DEX@1 to DEX@2, even when the source and target word orders were the same.

4.4. Analysis

This section analyzes the parsed outputs of DEX to understand what kind of structures the model has transferred across languages. More specifically, we analyze the length and type of constituents that could be correctly parsed by our cross-lingual model. For simplicity, we assume that we can choose the best source language for each target language. Therefore, only the DEX@1 model will be analyzed in this section.

4.4.1 Constituent Length

Figure 4.3 presents the performance of DEX in terms of constituent length. We use the supervised delexicalized model, DEX Sup, to define the upper-bound parsing performance of each language as the dash lines in each plot. The overall tendency of the DEX is described in the first plot, which shows the average performance of all languages. The values of unlabeled recall for DEX and DEX Sup

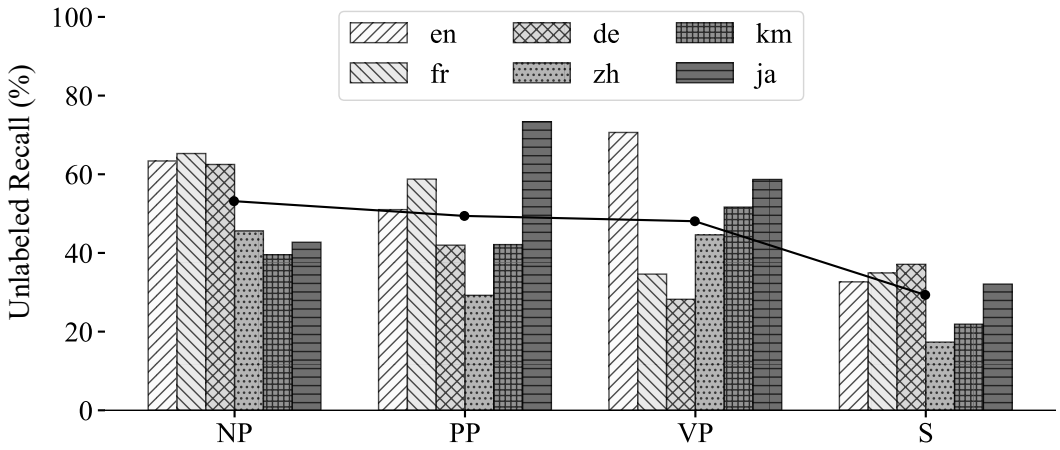


Figure 4.4: The unlabeled recall for all languages with respect to their constituent type. Each dot represents an average result for each constituent type.

reduce as constituent length increases. This is reasonable that long constituents tend to be more difficult to be parsed because the patterns of the long constituents are diverse and less frequent in the corpus. Besides, the gap between the DEX and DEX Sup models becomes gradually wider as constituent length increases. This might be because, for the long constituents, similar POS sequences between two languages were rare compared to the short constituents. The other plots in Figure 4.3 further present the parsing performance for each target language. The performance of DEX on short constituents is relatively high and even closed to each upper-bound performance for English, French, German, and Japanese. However, the tendency is less obvious for the Chinese, Khmer, and Myanmar. Additionally, the supervised performance is high for English and German regardless the length of the constituent, which results in a large gap in the performance between DEX and DEX Sup on the long constituents. This might be because the training treebanks for English and German are large compared to the other languages.

Table 4.6: Constituent labels mapping table

Labels	**Languages**					
	English	**French**	**German**	**Chinese**	**Khmer**	**Japanese**
NP	NAC, NX, PRN, WHNP	NC	PN, CNP	CLP, DNP, PRN	-	BASENP
VP	-	VN, VPinf, VPpart	CVP	DVP, VCD, VCP, VSB VNV, VPT, VRD	-	-
S	SBAR, SBARQ, SINV, SQ, RRC	Sint, Ssub, Srel	CS	IP, CP	-	SBAR
PP	WHPP	-	CPP	LCP	-	-

4.4.2 Constituent Types

This section analyzes the types of constituents induced by the DEX model. Because our experimental data are annotated using different sets of constituent labels, we map their labels sets to a general shared label set. For the shared label set, we choose only four major types of constituents: NP, VP, S, and PP. We map the constituent labels of each language according to Table 4.6. Of the other minority labels, those that are not in the table are ignored in this analysis. Unfortunately, we are not able to map the label set of Myanmar because only two labels, NOUN and VERB, are used in the Myanmar data. Therefore, we exclude the Myanmar language from this analysis.

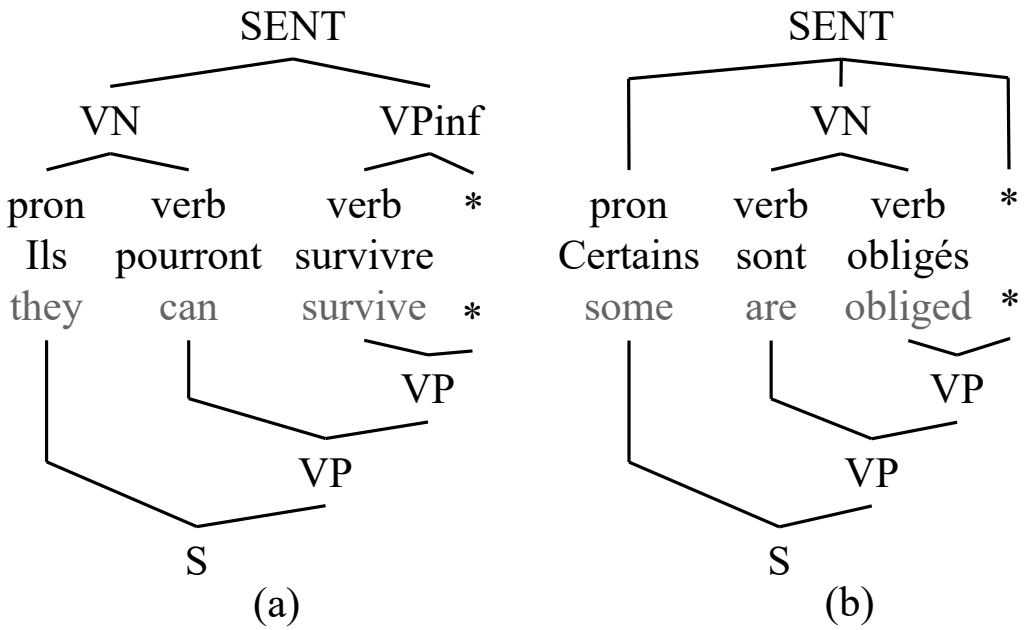


Figure 4.5: Partial tree examples for French. The upper structures are the ground-truth examples and lower structures are the induced examples. The words in gray are the English glosses of the French words. For remarks, VN and VPinf are mapped to VP for the analysis; and "Ils" in (a) is clitic originally tagged as "CLS".

Figure 4.4 shows the overall performance of the DEX model on each type of constituent. The constituent types are sorted in descending order by their averaged unlabeled recall values. The average result shows that the most difficult type of constituent for the DEX model was S and the easiest one was NP. For VP and PP, the accuracy of DEX was also pretty high compared with S. We might conclude here that it is because NP and PP are the short constituents and S is the long constituent according to our previous analysis on the length of constituent that the long constituents are more difficult than the short constituents. However, even though the VP constituent is commonly long, the model performance on this constituent was relatively high compared to the S constituent. Eventually, the performance regarding the constituent type depends on which source-target language pair to be transferred as the bar chart in Figure 4.4. For example, the performance on PP was low for Chinese while it was high for the other

languages. We find that the annotation of both preposition and postposition is equally common in Chinese but that of the postposition is rare in its source language, English. Interestingly, for French, the accuracy for VP was low while its source language is English and English-French is known as a similar language pair. For this, we find that some verbal annotations are different from those of English, for example, the clitic pronoun and deep function annotations. Specifically, there is a distinction between clitic pronouns and other pronouns and their structural annotations are different. For instance, as the upper structures in Figure 4.5, the clitic, "Ils" in (a), is combined with its following verb, "pourront", whereas the pronoun, "Certains" in (b), is considered as the subject of the sentence. In universal POS tags, clitic is annotated with the same tag as the other pronouns. As the result, their structures parsed by the DEX model are the same as the lower structures in Figure 4.5. In addition, as in (b), the objects or arguments are placed outside the verbal constituent in the ground-truth structure but inside the verbal constituent in the parsed structure. All these indicate the complex verbal structure of French constituency data. Such differences between ground-truth and parsed structure are reasonable.

4.5. Discussion

Our results indicate that a simple cross-lingual delexicalized model can achieve relatively high performance compared with all the baseline models if the best source language can be determined. the best source language can be accurately selected using current measurements, such as KLcpos3, precomputed syntax distance, or using a small treebank. However, we assume that similar languages exist in the source language pool but we cannot detect their absence. For instance, in our experiment, the word order of German is fundamentally different from that of other languages in the pool. Therefore, future work to solve this problem will be very important for low-resource settings such as source data transformation and the detection of the absence of similar languages in the pool.

Finally, since the performance of the fully unsupervised models, NPCFG and CPCFG, are impressive, the combination of both cross-lingual and unsupervised models is considered an interesting direction for future work of low-resource con-

stituency parsing.

4.6. Summary

This work investigated the performance of cross-lingual transfer, particularly delexicalized model transfer. We compared the performance of model transfer with state-of-the-art grammar induction models. We have confirmed that model transfer can achieve high performance if the best source languages are carefully selected. We have compared three methods of source selection and shown that using a small treebank of target languages for selection is highly accurate.

Additionally, we have shown that the parsing performance of source language has been reduced by delexicalization. Therefore, the following chapter will investigate the use of lexicalized cross-lingual embedding models but without using any bilingual resource. We will also investigate multi-source transfer scenarios to leverage all source-language parsing.

Chapter 5

Multi-Source Cross-Lingual Constituency Parsing

This chapter investigates the use of pretrained multilingual language models for the cross-lingual constituency parsing, and multi-source transfer scenario to leverage all source-languages parsing. Unlike the single-source transfer in Chapter 4, the multi-source transfer scenario combines treebanks of multiple languages together to train a multilingual parser that can be later used for any unseen language.

However, for constituency parsing, training a multilingual parser has two main issues that must be considered. First, the source languages can produce diverse word orders—for instance, different *subject-verb-object* or *noun-adjective* orders. These language properties can be simply identified using existing typology databases, e.g., WALS or SSWL. It is intuitive that these language properties can be used to guide a multilingual parser to share corresponding model parameters among similar languages [29,30,66,96]. For cross-lingual transfer, the typological features could hurt performance [29], and an effective integration technique is required [30]. Inspired by this, we investigate the usefulness of typological features for cross-lingual constituency parsing and propose a training strategy to generalize the cross-lingual capability of the model using smooth sampling and random dropout. The second issue is that even though constituency structure is universal, the design of a label set is language-specific. For dependency structures, this problem has inspired the creation of the Universal Dependency project [17]. The syntactic label sets of constituency structure vary across languages—for instance, very few labels are shared and even labels for the same syntactic category may be different across languages. This increases the complexity of the multi-source transfer. Therefore, we propose to preprocess the constituency treebanks to universalize the multilingual parsing model. In summary, to address both issues,

we propose 1) typological feature integration for model generalization on unseen languages (Section 5.3.1), and 2) a treebank preprocessing step is proposed to reduce the complexity of cross-lingual structural prediction (Section 5.3.2).

5.1. Related Works

In multi-source transfer, task-specific knowledge of multiple source languages is combined and jointly transferred to an unseen or zero-shot language. This combination can be categorized according to three levels [54], that is, the level of treebanks [29, 30, 58, 66], model parameters [67, 68], or parse outputs [63, 64]. This work focuses on the treebank level, that is, treebank concatenation and, unlike previous studies, we study a more sophisticated structure, constituency treebanks, which simultaneously contain diverse syntactic labels across multiple source languages.

Typological features are a valuable resource for multi-source transfer where source languages have diverse structures, and they have been used specifically for sharing the parameters of non-neural [96, 97, 105] and neural [29, 30, 66] models. Following the same motivation, we also investigate the usefulness of typological features for a multilingual constituent parser and propose a training strategy that generalizes the model for zero-shot languages. Specifically, we integrate typological features into the self-attentive constituency parser [51].

Our work is similar to that of Kitaev et al. [106] who investigated the multilingualism of the self-attentive constituency parser [51] using the pretrained multilingual language model. However, our work differs from theirs such that we focus on zero-shot performance. In addition, we propose to preprocess the concatenated treebanks and integrate typological features for better zero-shot performance. We also extend the sampling technique that Kitaev et al. [106] use by constraining the minimum size of each treebank.

5.2. The Self-Attentive Parser

The basis of our model (Fig. 5.1b) follows the self-attention-based encoder-decoder architecture of Kitaev et al. [51]. Specifically, the encoder consists of word embedding and self-attention layers to produce the contextual presentation

for each word. At the decoder side, all possible spans are extracted and each span (i, j) is represented by a hidden vector $v_{i,j}$ that is constructed by subtracting the representations associated with the start and end of the span. Then, each span (i, j) is assigned a labeling score $s(i, j, \cdot)$ by an MLP span classifier as

$$s(i, j, \cdot) = W_2 g(f(W_1 v_{i,j} + c_1)) + c_2, \tag{5.1}$$

where W_* and c_* are the weight and bias, respectively; f and g are the layer normalization and ReLU ("Re"ctified "L"inear "U"nit) activation function, respectively, as shown in Figure 5.1c. For each sentence, the constituency structure T is represented by a set of labeled spans $\{(i_t, j_t, l_t) : t = 1, \ldots, |T|\}$ where the score of T is

$$s(T) = \sum_{(i,j,l) \in T} s(i, j, l). \tag{5.2}$$

At test time, the optimal structure can be obtained using a CKY-style inference algorithm. For training, the model is optimized using a max-margin objective function, the details of which can be found in Kitaev et al. [51]. In addition, the parser's hyperparameters are unchanged from Kitaev et al. [51].

To perform cross-lingual parsing, an external pretrained multilingual language model must be used and simply take the place of the word embedding layer. Because the model is trained on sub-words, only the last sub-word unit of the corresponding token is used to represent a word. In this experiment, we use a recent multilingual language model, i.e. XLM-RoBERTa-Large* [107].

*This is a variant of XLM-RoBERTa, which has a larger parameters set.

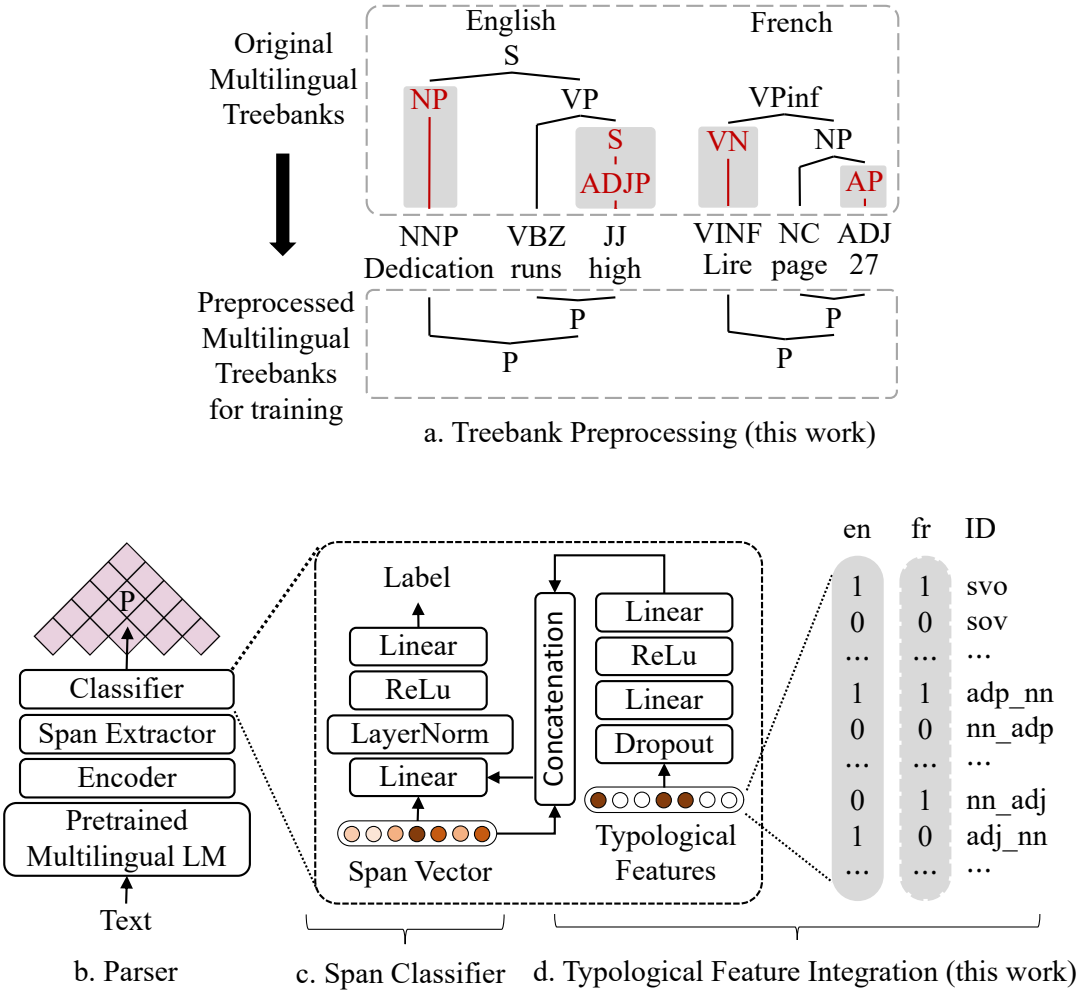


Figure 5.1: Overall architecture of our parser. The multilingual treebanks are preprocessed (a) before training the parser (b). A span classifier (c) is also integrated with a feature extractor (d) for binary typological vectors, as shown in the right-most example.

5.3. Proposed Methods

In this section, we present our multilingual constituency parser for cross-lingual transfer. Overall, our approach consists of treebank preprocessing and typological feature integration with smooth sampling and random dropout.

5.3.1 Typological Feature Integration

A typology database is a valuable resource that represents various aspects of languages. Recent `lang2vec` [99] provides an interface to represent languages as binary vectors of typological features. Inspired by the recent work of Ustun et al. [30], we also integrate typological features f (TF) into our model to guide the multilingual model's sharing of the structural knowledge among similar languages, as Figure 5.1d shows. We use simple feature concatenation to integrate typological features into the span classifier. Like Ustun et al. [30], we embed binary typological vectors using two linear layers and a ReLU activation function g, and further apply random dropout over the binary typological vectors as

$$f' = M_2 g(M_1 dropout(f) + z_1) + z_2. \tag{5.3}$$

We then concatenate f' with each span vector, $v_{i,j}$, which modifies Equation 5.1 as

$$s(i, j, \cdot) = W_2 g(f(W_1[v_{i,j}, f'] + c_1)) + c_2. \tag{5.4}$$

Dropout is applied directly to the binary features because, during training, typological features only vary with respect to the number of source languages, and each feature is only helpful in the context of other features, which is known as co-adaptation [108]. Therefore, for a zero-shot language, without dropout, the model would not be able to extract individual features in a new feature context, which can be prevented using simple random dropout [108]. Like Hinton et al. [108], we drop 50% of the features during training.

The number of multilingual treebanks commonly differs, and high-resource languages tend to be over-represented during training. Similar to the exponential smoothing in Kitaev et al. [106], at each epoch, we sample d^a examples from each

language, where d is the size of each language treebank and a is a hyperparameter. Unlike Kitaev et al. [106], we use $a = 0.95$ because the size of each treebank is not as large as the unlabeled corpora. We also constrain the smoothed number of examples as $d^a > m$, where m is the smallest treebank size in the source-language pool. We call this approach "smooth sampling."

For the typological features, we combine the syntax features of WALS [109] or SSWL [110]†. We only select the relevant features such as 81A, 82A, 83A, 85A, 87A, 88A, 89A, 90A, 144A, and other unknown ID features as in Table 5.1. In addition, we exclude the morphological features, which contain the word *prefix* or *suffix*, and the missing features of any source language. For zero-shot languages, the missing features are set to zero. After that, we further automatically remove unnecessary features that are repeated for all source languages. Like Ustun et al. [30], we set the hidden and output layer of our TF to 10 and 32, respectively.

5.3.2 Treebank Preprocessing

When combining multiple constituency treebanks, many syntactic disagreements among those treebanks could occur. One obvious issue is the difference in their syntactic labels. We observed that high-resource languages tend to have more diverse labels, whereas low-resource languages use a much smaller label set; for instance, Myanmar and Khmer have five and six labels, respectively, whereas English has 26. Moreover, label symbols for each treebank are very language-specific; for example, French and English, which have large label sets, only share two labels. Additionally, unary constituents that can be regarded as stacked labels of their child's labels might be common in some treebanks. We found that the number of stacks differs across the existing treebanks which is also problematic for the cross-lingual transfer evaluation.

Therefore, to solve this diversity, we propose a treebank preprocessing (TP) step when combining treebanks of multiple languages together. Firstly, we remove any non-terminal span that has length or number of children less than two, which is a unary span $(i, j) \in T$ where $j - i < 2$. After that, we mask the labels of

†These features can be obtained using `lang2vec` by passing a `syntax_wals+syntax_sswl.` argument.

Name	Feature ID
81A	SVO, SOV, VSO, VOS, OVS, OSV
82A	SUBJECT_BEFORE_VERB SUBJECT_AFTER_VERB
83A	OBJECT_AFTER_VERB OBJECT_BEFORE_VERB
85A	ADPOSITION_BEFORE_NOUN ADPOSITION_AFTER_NOUN
87A	ADJECTIVE_BEFORE_NOUN ADJECTIVE_AFTER_NOUN
8A	DEMONSTRATIVE_WORD_BEFORE_NOUN DEMONSTRATIVE_WORD_AFTER_NOUN
89A	NUMERAL_BEFORE_NOUN NUMERAL_AFTER_NOUN
90A	RELATIVE_BEFORE_NOUN RELATIVE_AFTER_NOUN RELATIVE_AROUND_NOUN
144A	NEGATIVE_WORD_BEFORE_VERB NEGATIVE_WORD_AFTER_VERB NEGATIVE_WORD_INITIAL NEGATIVE_WORD_FINAL NEGATIVE_WORD_ADJACENT_BEFORE_VERB NEGATIVE_WORD_ADJACENT_AFTER_VERB
Others	SUBJECT_BEFORE_OBJECT SUBJECT_AFTER_OBJECT POSSESSOR_BEFORE_NOUN POSSESSOR_AFTER_NOUN DEGREE_WORD_BEFORE_ADJECTIVE DEGREE_WORD_AFTER_ADJECTIVE SUBORDINATOR_WORD_BEFORE_CLAUSE SUBORDINATOR_WORD_AFTER_CLAUSE

Table 5.1: List of typological features.

all the remaining non-terminal spans with an unified symbol, e.g., "P" as in the example in Figure 5.1a. As a result, our label classifier is simplified to only detect the span as a span or non-span.

5.4. Experiment

5.4.1 Setting

Dataset: The evaluation was performed on 14 languages: English from the Penn Treebank [100]; Chinese from the Chinese Penn Treebank 5.1 [101]; Japanese, Khmer, and Myanmar from the Asian Language Treebank [21]; and Arabic, Basque, French, German, Hebrew, Hungarian, Korean, Polish, and Swedish from the SPMRL 2013 shared task [103]. The standard splits of each treebank were applied to prepare the training, validation, and test datasets. We grouped the languages into high- and low-resource (zero-shot) languages based on their amount of data; those with fewer than 10k samples were treated as low-resource languages (Khmer, Hungarian, Basque, Polish, Swedish, and Hebrew). We trained a multilingual model on the high-resource languages and evaluated the cross-lingual parsing on the low-resource languages. Note that Khmer and Myanmar scripts have no word boundaries, so we simply use their gold segmented long token[‡] for this experiment. We observe that XLM-RoBERTa-Large's tokenizer produces reasonable sub-words for Khmer and Myanmar's long tokens, even when the tokenizer was trained using SentencePiece for these two languages. Table 5.2 presents detailed data statistics for each language.

Baselines: We trained two baselines, single- and multi-source models. For the single-source model, we trained the parser for each high-resource language and then selected the best model based on its parsing accuracy on the oracle test set (S_{best}) of the low-resource language or used the precomputed syntactic distance [99] (S_{dist}). For the same-value syntactic distance, we further weighted each source language based on the size of its corresponding training data. For the multi-source baseline, a multilingual parser (M_{base}) was trained on concatenated

[‡]Khmer and Myanmar written scripts can be segmented into morphemes (short tokens) or at compound level (long tokens) [23]

Code	Language	Train	Valid	Test
de	German	40,472	5,000	5,000
en	English	39,832	1,700	2,416
ko	Korean	23,010	2,066	2,287
my	Myanmar	18,088	1,000	1,018
zh	Chinese	17,544	352	348
ja	Japanese	17,204	953	931
ar	Arabic	15,762	1,985	1,959
fr	French	14,759	1,235	2,541
km	Khmer	8,788	510	654
hu	Hungarian	8,146	1,051	1,009
eu	Basque	7,577	948	946
pl	Polish	6,578	821	822
sv	Swedish	5,000	494	666
he	Hebrew	5,000	500	716

Table 5.2: Data statistics. The numbers refer to numbers of sentences where upper languages are high-resource languages and lower for low-resource languages.

treebanks without the treebank preprocessing step or typological features.

Evaluation: The outputs of all models were post-processed including removal of trivial spans (width-one/unary and sentence-level spans). Unlike the evaluation in Chapter 4, punctuation marks were kept in this chapter. We calculated the unlabeled F1 measure to evaluate cross-lingual performance. All following F1 values refer to the unlabeled F1 for simplicity. Worth noting that the removal of trivial spans and the evaluation without labels is equivalent to applying our treebank preprocessing step.

5.4.2 Results

As shown in Table 5.3, the performance of single-source transfer was very high, especially when the best source language can be accurately detected. Unfortunately, the precomputed syntactic distance is not enough to choose the best

Model	km	hg	eu	pl	sv	he	avg
S_{best}	70.0	64.7	33.2	**72.8**	**74.8**	77.1	64.5
S_{dist}	70.0	31.2	27.2	**72.8**	**74.8**	71.3	55.5
M_{base}	55.5	68.6	27.3	65.6	67.8	80.5	61.9
TP_{ours}	69.0	73.9	34.7	67.9	73.1	81.6	66.2
$TP+TF_{ours}$	**71.8**	**74.7**	**35.8**	68.3	73.8	**82.2**	**67.0**

Table 5.3: Main results in unlabeled F1. The best F1 for each row is highlighted in bold text.

source language; in the results, it failed in three out of six cases. The alternative to source selection is to train a multilingual parser. Interestingly, even the straightforward treebank concatenation M_{base} has a competitive performance when compared with the single-source transfer. The results further show that the treebank preprocessing step is essential when training a multilingual constituency parser for zero-shot languages, where the improvement over M_{base} is 4.3 in average F1. This result suggests that reducing the complexity of the structure improves cross-lingual performance. In addition to the treebank preprocessing step, our integration of typological features constantly improves cross-lingual performance.

5.4.3 Analysis

Effect of Smooth Sampling and Dropout

This section analyzes the effect of the smooth sampling and dropout as in Figure 5.2. The analysis shows that the straightforward integration of typological features yields smaller improvements or hurts the performance for some zero-shot languages, indicating the effectiveness of our smooth sampling and dropout, which generalize the typology-guided cross-lingual parser for zero-shot languages. We additionally observe that the combination of both smooth sampling and dropout is the best configuration for the cross-lingual parsing.

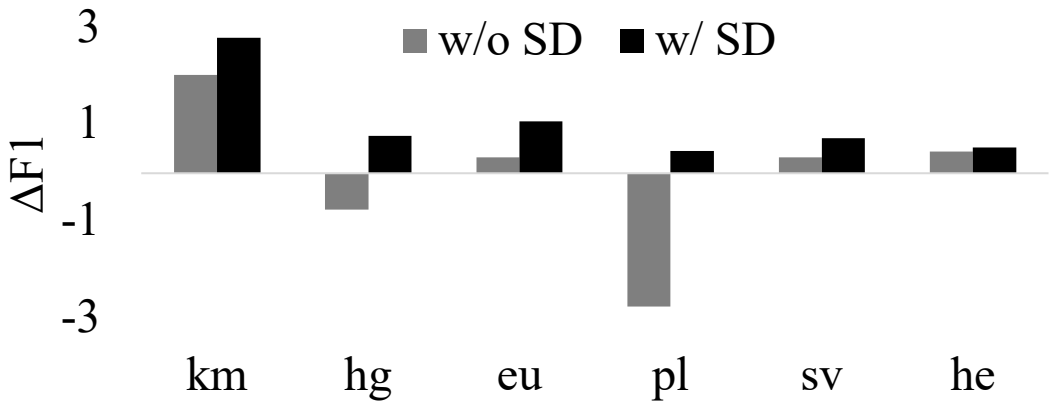


Figure 5.2: Improvements in the F1 of TP+TF over TP model with or without SD. SD refers to **S**mooth sampling and random **D**ropout.

Effect of Pretrained Multilingual Language Models

We have demonstrated the high performance of cross-lingual constituency parsing using the pretrained multilingual language model—XLM-RoBERTa-Large (XLM-R-L), of which performance is up to 70 F1 for most target languages. Since the cross-lingual model is based on its cross-lingual input representation, this section further compares the use of the pretrained multilingual language model and the delexicalized model (DEX) that we presented in Chapter 4. In addition, we also analyze how other pretrained multilingual language models, i.e., mBERT [111] and the smaller variant of the XLM-RoBERTa (XLM-R) [107], be effective for cross-lingual constituency parsing. To compare with DEX, we will only analyze for seven languages, that is, English, Chinese, French, Khmer, German, Japanese, and Myanmar, which were studied in Chapter 4. It is also because the POS-tags of the other languages are difficult to be projected to the universal POS-tags. Noting that the F1 performance of the delexicalized model will differ from that of in Chapter 4 because the punctuation is included in this evaluation.

Table 5.4 presents the comparison in terms of the supervised and cross-lingual performance. The supervised performance is based on the evaluation of the parser of each target language. The cross-lingual performance is based on the evaluation of the single source transfer scenario with S_{dist} or S_{best} selection approach, of which source languages are among the seven aforementioned languages. As a

Model	en	zh	fr	km	de	ja	my	avg
Supervised parsing								
DEX	84.5	79.2	73.1	71.3	78.7	70.3	83.7	77.2
mBERT	95.1	93.1	94.1	46.5	83.3	88.9	94.3	85.0
XLM-R	95.7	92.5	95.2	89.8	84.0	89.7	95.2	91.7
XLM-R-L	**96.1**	**92.8**	**95.9**	**90.6**	**84.3**	**90.5**	**95.9**	**92.3**
Cross-lingual transfer with S_{dist}								
DEX	40.4	58.9	47.0	37.2	50.8	42.9	35.8	44.7
mBERT	54.1	71.5	51.5	18.4	57.5	62.3	50.4	52.3
XLM-R	**53.8**	**75.1**	55.1	66.5	**61.1**	62.6	48.2	**60.3**
XLM-R-L	50.8	66.7	**58.4**	**70.0**	54.5	**63.9**	**54.5**	59.8
Cross-lingual transfer with S_{best}								
DEX	52.1	58.9	47.0	38.9	50.8	42.9	40.8	47.3
mBERT	63.7	71.5	51.5	22.3	57.5	62.3	51.7	54.4
XLM-R	68.8	**75.1**	55.1	66.5	**61.1**	62.6	48.2	62.5
XLM-R-L	**77.1**	68.4	**60.1**	**70.0**	54.5	**63.9**	**57.2**	**64.5**

Table 5.4: Comparison of the cross-lingual input representation.

result, all of the pretrained multilingual language models are better than DEX in both supervised and cross-lingual performance, which indicates that the pretrained models are highly beneficial for cross-lingual transfer because there is no requirement of any bilingual information or POS-tag projection. Another advantage is from their diverse lexical information and sub-word processing that preserve or boost the source-language parsing performance. However, the performance of the mBERT for Khmer is very low that is because the mBERT's tokenizer chunk the Khmer word into characters, which has been fixed in the XLM-RoBERTa model by exponential-smoothing-based sampling. In addition, the XLM-RoBERTa models tend to perform better than mBERT in most cases and the performance of the XLM-R-L and XLM-R are comparable.

5.5. Summary

We demonstrated the strong ability of recent pretrained multilingual language models for cross-lingual constituency parsing. This result will serve as a new benchmark for future cross-lingual constituency parsing. Moreover, we found that our treebank preprocessing step is crucial when training multilingual treebanks with diverse label sets. In addition, our typological feature integration with dropout and smooth sampling generalizes and improves the model for zero-shot languages. Because we integrated typological features into the span classifier using a simple concatenation approach, more advanced techniques—for instance, a parameter generator [30]—with our dropout and smooth sampling could be studied in the future.

Chapter 6

Conclusions

6.1. Summary

This thesis addressed the fundamental problems of the NLP systems, that is, tokenization, POS tagging, and constituency parsing for the Khmer language. Preliminary, we developed a corpus and investigated the tokenization and POS tagging for the Khmer language as in Chapter 3. After that, we studied cross-lingual transfer for constituency parsing as in Chapters 4 and 5.

In Chapter 3, we proposed a tokenized and POS-tagged Khmer corpus that contains annotation for the grammaticalization and compounding. The corpus consists of $20,000$ sentences. The manual annotation processing started in 2016, and the corpus was the largest morphological annotated Khmer dataset when this work was completed in 2020. Khmer is extremely analytic such that its syntactic information is overwhelmingly afforded by word order with abundant grammaticalization phenomena. Therefore, the tokenization and POS tagging processing for Khmer are considerably ambiguous. We investigated the automatic tokenization and POS tagging based on our annotated data using four standard approaches of SVM, CRF, LSTM-RNN, and LSTM-CRF. Our experimental 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. 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.

In Chapter 4, we proposed a cross-lingual constituency parser by delexicalization, which outperformed the fully-unsupervised baselines in most cases. We

confirmed that the model transfer can achieve high performance if the best source languages are carefully selected. We have compared three source selection methods and shown that using a small treebank of target languages for selection is highly accurate. In addition, we observed that the delexicalization sacrificed the supervised parsing performance for each source language that was then addressed in Chapter 5 by preserving the lexical information.

In Chapter 5, we examined the performance of a multi-source cross-lingual constituency parsing model and proposed a treebank preprocessing step and typological features integration to generalize the cross-lingual performance of the model. Specifically, we demonstrated the strong ability of recent pretrained multilingual language models for lexicalized cross-lingual constituency parsing even without using bilingual information. We investigated multi-source transfer, which does not require source-selection technique and in some way outperform best-selected single source transfer. Lastly, we showed that our treebank preprocessing step is crucial when training multilingual treebanks with diverse label sets. In addition, our typological feature integration with dropout and smooth sampling generalizes and improves the model for zero-shot languages. Lastly, we have also shown the state-of-the-art supervised and zero-shot performance for the Khmer language in this chapter. Specifically, the supervised performance is up to 90.6 and the zero-shot performance based on our model is up to 71.8.

6.2. Future Directions

On top of our tokenized and POS tagged corpus, the syntactic annotation has been completed as a final treebank for Khmer and is going to release in near future. We plan to conduct experiments on the ultimate joint processing of tokenization, POS-tagging, and syntactic parsing on our final treebank. Additionally, as the experiments showed, a proper noun dictionary is necessary. Therefore, like Mon et al. [112], we also plan to develop a dictionary of borrowed words in Khmer after completing the final treebank.

For cross-lingual parsing, there are rooms to improve our approach that we plan to study in the future. Specifically, firstly, even we have shown that the concatenation technique for typological features integration yield a good improvement,

more advanced techniques—for instance, a parameter generator [30]—with our dropout and smooth sampling will be studied in the future. Secondly, as the typological features are valuable and useful for the cross-lingual transfer, further analysis of each feature's contribution to the approach will provide more insight and allow to select more relevant features for transferring. This will also be studied in our future work.

Additionally, we are also interested to fine-tune our model in Chapter 5 using each target language's treebank. However, our preliminary experiment cannot observe any obvious improvement using a simple fine-tuning technique that simply uses our model for parameter initialization. Intuitively, surpassing the performance of the supervised model is difficult for two reasons. First, the supervised model leveraged the pretrained language model that has overcome the unknown vocabulary problems. Second, structures in both train and test sets are very similar in that most structures in the test set might be covered in the training set as their accuracies are already higher than 90% in most cases as in Table 5.4. Therefore, an attentive fine-tuning experiment including setup, analysis, and discussion, is necessary, which will be studied in the future as well.

Furthermore, we also would like to discuss the application of our works for future NLP systems. As we have developed a tokenized and POS tagged corpus for Khmer morphemes and words our corpus will be used for the basic step—tokenization—of the NLP tasks for Khmer, e.g., the multilingual NLP tasks that are being studied for more than a hundred languages. Based on our knowledge, there are language modeling [107], machine translation [113], text summarization [114], named entity recognition [115], and question answering [116] tasks. None of these tasks tokenize texts for Khmer into words or morphemes. Instead, the Khmer texts were chunked using sentencepiece approach [107, 113], or into characters sequence [115]. However, the experiment without a tokenizer is difficult to be conducted [114]. Therefore, our corpus will facilitate future NLP tasks or will be used to improve the system performance. Besides, we plan to analyze the performance of the various NLP tasks when using the morpheme or word as the input unit. In addition, we are also interested to explore the usefulness of the POS-tag information for machine translation for Khmer, which is inspired by the previous works [5, 7].

The syntax is essential for the NLP systems because it represents the language interpretation behavior of humans [117]. Its contribution to the NLP systems has been demonstrated in various studies, e.g., syntax-based language modeling [10], text summarization [16], and machine translation [11, 14, 15]; and the syntax information tends to be more useful for low-resource languages [11, 13–15]. However, the syntax of English was commonly used for these studies due to the high quality of the English treebanks and it can be more interesting to investigate the use of the syntax of the low-resource languages. Besides we have shown that our zero-shot parser can induce a good pseudo syntax, the future syntax-based NLP systems should be able to benefit our zero-shot parser. Certainly, our zero-shot parser may not perform well for all the languages because of the limited number of source languages for training the model. However, it is possible to validate the quality of the pseudo syntax based on the performance of the downstream tasks and to improve the parser performance via a multi-task approach [10], which are listed in our plans.

Acknowledgements

I would like to express my deep and sincere gratitude to Professor Satoshi Nakamura and Associate Professor Katsuhito Sudoh for providing invaluable guidance throughout my doctoral research. They are supportive and inspiring. I have learned a lot about how to do research and write good scientific papers from them. I also would like to extend my gratitude to Research Associate Professor Sakriani Sakti, Affiliate Associate Professor Koichiro Yoshino, and Affiliate Associate Professor Keiji Yasuda and all AHC lab staff for their insightful suggestion during group meetings and seminars. Furthermore, I would like to thank Professor Yuji Matsumoto and Professor Taro Watanabe for their advice and comments on my Ph.D. research.

I also would like to take this opportunity to thank Dr. Eiichiro Sumita and Dr. Masao Utiyama for offering me a chance to do research in Japan. They encouraged and supported me to pursue my Ph.D. at NAIST and gave me a lot of great advice during my Ph.D. research. Other big thanks to Dr. Chenchen Ding for his guidance and helps especially to write professional scientific papers. Additionally, I want to thank my colleagues, Dr. Rui Wang, Dr. Benjamin Marie, Dr. Raphael Rubino, Dr. Raj Dabre, Dr. Atsushi Fujita, Dr. Andrew Finch, Dr. Hideki Tanaka for the discussion and comments. I also thank Mr. Shoji Shiomi and Mr. Takashi Maruno for their technical support, and all other colleagues for their help during my stay at NICT.

I also want to thank our NAIST lab assistant Ms. Manami Matsuda, our NICT lab assistant Ms. Chikako Tamasaki, Ms. Mari Oku, Ms. Yumiko Nishiyashiki, and Ms. Tetsuko Yamaji for their facilitation on a wide range of tasks, e.g. administrative works and many things about life in Japan as a foreigner.

Moreover, I also thank Dr. Hiroaki Kato and Ms. Yayoi Tominaga for their help since my first day in Japan. They showed me a lot of things about Japanese culture and life. At the same time, thanks to all the members in NICT speech lab, especially Dr. Takuma Okamoto for always inviting me to the party and Dr. Masakiyo Fujimoto for his great cooking.

Additionally, I want to thank all my friends in Japan, Cambodia, and other countries for their friendship. I also want to give a special thanks to Ms. Seakmouy Vuthy for her love and help with life in Japan. Finally, I really thank all my family members for their love, support, and encouragement.

References

[1] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to Japanese morphological analysis. In *Proc. of EMNLP*, pages 230–237, 2004.

[2] Chenchen Ding, Hnin Thu Zar Aye, Win Pa Pa, Khin Thandar Nwet, Khin Mar Soe, Masao Utiyama, and Eiichiro Sumita. Towards Burmese (Myanmar) morphological analysis: syllable-based tokenization and part-of-speech tagging. *ACM Trans. Asian Low-Resour. Lang. Info. Process.*, 19(1):5, 2019.

[3] Philipp Koehn and Hieu Hoang. Factored translation models. In *Proc. of EMNLP-CoNLL)*, pages 868–876, 2007.

[4] Hour Kaing, Chenchen Ding, Masao Utiyama, Eiichiro Sumita, and Vichet Chea. Improving english-to-khmer statistical machine translation using part-of-speech information. In *Proc. of KNLP*, 2016.

[5] Rico Sennrich and Barry Haddow. Linguistic input features improve neural machine translation. In *Proc. of WMT*, pages 83–91, 2016.

[6] Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, Tiejun Zhao, Muyun Yang, and Hai Zhao. Towards more diverse input representation for neural machine translation. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 28:1586–1597, 2020.

[7] Abhisek Chakrabarty, Raj Dabre, Chenchen Ding, Masao Utiyama, and Eiichiro Sumita. Improving low-resource nmt through relevance based linguistic features incorporation. In *Proc. of COLING*, pages 4263–4274, 2020.

[8] Fabian Hommel, Philipp Cimiano, Matthias Orlikowski, and Matthias Hartung. Extending neural question answering with linguistic input features. In *Proc. of SemDeep)*, pages 31–39, 2019.

[9] Courtney Mansfield, Ming Sun, Yuzong Liu, Ankur Gandhe, and Björn Hoffmeister. Neural text normalization with subword units. In *Proc. of NAACL)*, pages 190–196, 2019.

[10] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. Recurrent neural network grammars. In *Proc. of NAACL*, pages 199–209, 2016.

[11] Anna Currey and Kenneth Heafield. Incorporating source syntax into transformer-based neural machine translation. In *Proc. of WMT*, pages 24–33, 2019.

[12] Vighnesh Shiv and Chris Quirk. Novel positional encodings to enable tree-based transformers. *Advances in Neural Information Processing Systems*, 32:12081–12091, 2019.

[13] Chunpeng Ma, Akihiro Tamura, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. Syntax-based transformer for neural machine translation. *Journal of Natural Language Processing*, 27(2):445–466, 2020.

[14] Zuchao Li, Masao Utiyama, Eiichiro Sumita, and Hai Zhao. Unsupervised neural machine translation with universal grammar. In *Proc. of EMNLP*, pages 3249–3264, 2021.

[15] Colin McDonald and David Chiang. Syntax-based attention masking for neural machine translation. In *Proc. of NAACL*, pages 47–52, 2021.

[16] Jiacheng Xu and Greg Durrett. Neural extractive text summarization with syntactic compression. In *Proc. of EMNLP*, 2019.

[17] Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *Proc. of LREC*, pages 1659–1666, 2016.

[18] Narin Bi and Nguonly Taing. Khmer word segmentation based on bi-directional maximal matching for plaintext and microsoft word document. In *Proc. of APSIPA*, pages 1–9, 2014.

[19] Vichet Chea, Ye Kyaw Thu, Chenchen Ding, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. Khmer word segmentation using conditional random fields. In *Proc. of Khmer NLP*, pages 62–69, 2015.

[20] Chea Sok Huor, Top Rithy, Ros Pich Hemy, Vann Navy, Chin Chanthirith, and Chhoeun Tola. Word bigram vs orthographic syllable bigram in Khmer word segmentation. *PAN Localization Working Papers*, 2004.

[21] Hammam Riza, Michael Purwoadi, 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. Introduction of the Asian language treebank. In *Proc. of O-COCOSDA*, pages 1–6, 2016.

[22] Ye Kyaw Thu, Chea Vichet, and Sagisaka Yoshinori. Comparison of six POS tagging methods on 12k sentences Khmer language POS tagged corpus. In *Proc. of ONA*, 2017.

[23] Chenchen Ding, Masao Utiyama, and Eiichiro Sumita. 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):17, 2018.

[24] Chenchen Ding, Masao Utiyama, and Eiichiro Sumita. Tokenized and POS-Tagged Khmer Data of the Asian Language Treebank Project, July 2020.

[25] Madeline Elizabeth Ehrman and Kem Sos. *Contemporary Cambodian: grammatical sketch*, volume 8631. Foreign Service Institute, Department of State, 1972.

[26] Chunpeng Ma, Akihiro Tamura, Masao Utiyama, Tiejun Zhao, and Eiichiro Sumita. Forest-based neural machine translation. In *Proc. of ACL*, pages 1253–1263, 2018.

[27] Chunpeng Ma, Akihiro Tamura, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. Improving neural machine translation with neural syntactic distance. In *Proc. of NAACL*, pages 2032–2037, 2019.

[28] Wasi Uddin Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard H. Hovy, Kai-Wei Chang, and Nanyun Peng. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *Proc. of NAACL-HLT*, pages 2440–2452, 2019.

[29] Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A Smith. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444, 2016.

[30] Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. Udapter: Language adaptation for truly universal dependency parsing. In *Proc. of EMNLP*, pages 2302–2315, 2020.

[31] Noam Chomsky. *Aspects of the Theory of Syntax*, volume 11. MIT press, 2014.

[32] National Institute of Statistics Ministry of Planning. General population census of the kingdom of cambodia 2019. *Ministry of Planning, National Institute of Statistics*, 53:1–50, 2019.

[33] Joshua Horton, Makara Sok, Marc Durdin, and Rasmey Ty. Spoof-vulnerable rendering in khmer unicode implementations. In *Proc. of ACIS*, pages 177–180, 2017.

[34] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001.

[35] Graham Neubig and Shinsuke Mori. Word-based partial annotation for efficient corpus construction. In *Proc. of LREC*, 2010.

[36] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proc. of NAACL*, pages 213–220, 2003.

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

[38] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proc. of NAACL-HLT*, pages 260–270, 2016.

[39] Kenji Sagae and Alon Lavie. A classifier-based parser with linear run-time complexity. In *Proc. of IWPT*, pages 125–132, 2005.

[40] Yue Zhang and Stephen Clark. Transition-based parsing of the chinese treebank using a global discriminative model. In *Proc. of IWPT*, pages 162–171, 2009.

[41] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. Fast and accurate shift-reduce constituent parsing. In *Proc. of ACL*, pages 434–443, 2013.

[42] Taro Watanabe and Eiichiro Sumita. Transition-based neural constituent parsing. In *Proc. of ACL*, pages 1169–1179, 2015.

[43] James Cross and Liang Huang. Incremental parsing with minimal features using bi-directional lstm. In *Proc. of ACL*, pages 32–37, 2016.

[44] Nikita Kitaev and Dan Klein. Tetra-tagging: Word-synchronous parsing with linear-time inference. In *Proc. of ACL*, pages 6255–6261, 2020.

[45] Noam Chomsky. Formal properties of grammars. *Handbook of Math. Psychology*, 2:328–418, 1963.

[46] Michael Collins. Three generative, lexicalised models for statistical parsing. *Proc. of EACL*, pages 16–23, 1997.

[47] Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proc. of ACL*, pages 423–430, 2003.

[48] David Hall, Greg Durrett, and Dan Klein. Less grammar, more features. In *Proc. of ACL*, pages 228–237, 2014.

[49] Mitchell Stern, Jacob Andreas, and Dan Klein. A minimal span-based neural constituency parser. In *Proc. of ACL*, page 818827, 2017.

[50] David Gaddy, Mitchell Stern, and Dan Klein. What's going on in neural constituency parsers? an analysis. In *Proc. of NAACL*, page 9991010, 2018.

[51] Nikita Kitaev and Dan Klein. Constituency parsing with a self-attentive encoder. In *Proc. of ACL*, pages 2676–2686, 2018.

[52] Tadao Kasami. An efficient recognition and syntax-analysis algorithm for context-free languages. *Coordinated Science Laboratory Report no. R-257*, 1966.

[53] Daniel H Younger. Recognition and parsing of context-free languages in time n3. *Information and control*, 10(2):189–208, 1967.

[54] Ayan Das and Sudeshna Sarkar. A survey of the model transfer approaches to cross-lingual dependency parsing. *ACM Trans. Asian Low-Resour. Lang. Info. Process.*, 19(5):1–60, 2020.

[55] Wenbin Jiang, Qun Liu, and Yajuan Lü. Relaxed cross-lingual projection of constituent syntax. In *Proc. of EMNLP*, pages 1192–1201, 2011.

[56] Isao Goto, Masao Utiyama, Eiichiro Sumita, and Sadao Kurohashi. Preordering using a target-language parser via cross-language syntactic projection for statistical machine translation. *ACM Trans. Asian Low-Resour. Lang. Info. Process.*, 14(3):1–23, 2015.

[57] Daniel Zeman and Philip Resnik. Cross-language parser adaptation between related languages. In *Proc. of IJCNLP*, 2008.

[58] Ryan McDonald, Slav Petrov, and Keith B Hall. Multi-source transfer of delexicalized dependency parsers. In *Proc. of EMNLP*, pages 62–72, 2011.

[59] Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Proc. of NAACL-HLT*, pages 1599–1613, 2019.

[60] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168, 2013.

[61] Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *Proc. of ICLR*, 2018.

[62] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning bilingual word embeddings with (almost) no bilingual data. In *Proc. of ACL*, pages 451–462, 2017.

[63] Rudolf Rosa and Zdeněk Žabokrtský. KLcpos3-a language similarity measure for delexicalized parser transfer. In *Proc. of ACL-IJCNLP*, pages 243–249, 2015.

[64] Željko Agić. Cross-lingual parser selection for low-resource languages. In *Proc. of UDW*, pages 1–10, 2017.

[65] Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, et al. Choosing transfer languages for cross-lingual learning. In *Proc. of ACL*, page 31253135, 2019.

[66] Manon Scholivet, Franck Dary, Alexis Nasr, Benoit Favre, and Carlos Ramisch. Typological features for multilingual delexicalised dependency parsing. In *Proc. of NAACL*, pages 3919–3930, 2019.

[67] Shay B Cohen, Dipanjan Das, and Noah A Smith. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proc. of EMNLP*, pages 50–61, 2011.

[68] Anders Søgaard and Julie Wulff. An empirical etudy of non-lexical extensions to delexicalized transfer. In *Proc. of COLING*, pages 1181–1190, 2012.

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

[70] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[71] Jie Yang, Shuailong Liang, and Yue Zhang. Design challenges and misconceptions in neural sequence labeling. In *Proc. of COLING*, pages 3879–3889, 2018.

[72] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[73] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proc. of ACL*, pages 1064–1074, 2016.

[74] Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In *Proc. of ACL*, pages 1756–1765, 2017.

[75] Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. A unified character-based tagging framework for Chinese word segmentation. *ACM Trans. Asian Lang. Info. Process.*, 9(2):1–32, 2010.

[76] Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuan-Jing Huang. Long short-term memory neural networks for Chinese word segmentation. In *Proc. of EMNLP*, pages 1197–1206, 2015.

[77] Canasai Kruengkrai, Kiyotaka Uchimoto, Junichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara. An error-driven word-character hybrid model for joint Chinese word segmentation and POS tagging. In *Proc. of ACL-AFNLP*, pages 513–521, 2009.

[78] Sopheap Seng, Sethserey Sam, Laurent Besacier, Brigitte Bigi, and Eric Castelli. First broadcast news transcription system for Khmer language. In *Proc. of LREC*, pages 2658–2661, 2008.

[79] Chenda Nou and Wataru Kameyama. Transformation-based Khmer part-of-speech tagger. In *Proc. of ICAI*, pages 581–587, 2007.

[80] Chenda Nou and Wataru Kameyama. Hybrid approach for Khmer unknown word POS guessing. In *Proc. of IRI*, pages 215–220, 2007.

[81] Chenchen Ding, Sann Su Su Yee, Win Pa Pa, Khin Mar Soe, Masao Utiyama, and Eiichiro Sumita. A Burmese (Myanmar) treebank: guideline and analysis. *ACM Trans. Asian Low-Resour. Lang. Info. Process.*, 19(3):40, 2020.

[82] Chenchen Ding, Vichet Chea, Masao Utiyama, Eiichiro Sumita, Sethserey Sam, and Sopheap Seng. Statistical Khmer name romanization. In *Proc. of PACLING*, pages 179–190, 2017.

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

[84] 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. Dynet: the dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*, 2017.

[85] Diederik P Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *Proc. of ICLR*, 2015.

[86] Jie Yang and Yue Zhang. Ncrf++: an open-source neural sequence labeling toolkit. In *Proc. of ACL, System Demonstrations*, pages 74–79, 2018.

[87] Peter Sollich and Anders Krogh. Learning with ensembles: How overfitting can be useful. In *Proc. of NIPS*, pages 190–196, 1995.

[88] Alexander Clark. Unsupervised induction of stochastic context-free grammars using distributional clustering. In *Proc. of ConLL*, 2001.

[89] Dan Klein and Christopher D Manning. A generative constituent-context model for improved grammar induction. In *Proc. of ACL*, pages 128–135, 2002.

[90] Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron Courville. Neural language modeling by jointly learning syntax and lexicon. In *Proc. of ICLR*, 2018.

[91] Andrew Drozdov, Pat Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. Unsupervised latent tree induction with deep inside-outside recursive autoencoders. In *Proc. of NAACL*, pages 1129–1141, 2019.

[92] Yoon Kim, Alexander M Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. Unsupervised recurrent neural network grammars. *Proc. of NAACL*, 2019.

[93] Yoon Kim, Chris Dyer, and Alexander M Rush. Compound probabilistic context-free grammars for grammar induction. In *Proc. of ACL*, pages 2369–2385, 2019.

[94] Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara I. Cabezas, and Okan Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng.*, 11(3):311–325, 2005.

[95] Jörg Tiedemann, Zeljko Agic, and Joakim Nivre. Treebank translation for cross-lingual parser induction. In *Proc. of CoNLL*, pages 130–140, 2014.

[96] Tahira Naseem, Regina Barzilay, and Amir Globerson. Selective sharing for multilingual dependency parsing. In *Proc. of ACL*, page 629637, 2012.

[97] Oscar Täckström, Ryan McDonald, and Joakim Nivre. Target language adaptation of discriminative transfer parsers. In *Proc. of NAACL*, pages 1061–1071, 2013.

[98] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *Proc. of LREC*, pages 2089–2096, 2012.

[99] Patrick Littell, David R Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proc. of EACL*, pages 8–14, 2017.

[100] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330, 1993.

[101] Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207, 2005.

[102] Anne Abeillé, Lionel Clément, and Alexandra Kinyon. Building a treebank for French. In *Proc. of LREC*, 2000.

[103] Djamé Seddah, Sandra Kübler, and Reut Tsarfaty. Introducing the SPMRL 2014 shared task on parsing morphologically-rich languages. In *Proc. of SPMRL*, pages 103–109, 2014.

[104] Yanpeng Zhao. An empirical study of compound pcfgs. *https://github.com/zhaoyanpeng/cpcfg*, 2020.

[105] Yuan Zhang and Regina Barzilay. Hierarchical low-rank tensors for multilingual transfer parsing. In *Proc. of EMNLP*, pages 1857–1867, 2015.

[106] Nikita Kitaev, Steven Cao, and Dan Klein. Multilingual constituency parsing with self-attention and pre-training. In *Proc. of ACL*, pages 3499–3505, 2019.

[107] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proc. of ACL*, pages 8440–8451, 2020.

[108] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580. [cs.NE]*, 2012.

[109] Matthew S. Dryer and Martin Haspelmath. The world atlas of language structures online, 2013.

[110] Chris Collins and Richard Kayne. Syntactic structures of the worlds languages, 2011.

[111] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*, pages 4171–4186, 2018.

[112] Aye Myat Mon, Chenchen Ding, Hour Kaing, Khin Mar Soe, Masao Utiyama, and Eiichiro Sumita. A Myanmar (Burmese)-English named entity transliteration dictionary. In *Proc. of LREC*, pages 2980–2983, 2020.

[113] Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *arXiv preprint arXiv:2106.03193*, 2021.

[114] Daniel Varab and Natalie Schluter. Massivesumm: a very large-scale, very multilingual, news summarisation dataset. In *Proc. of EMNLP*, pages 10150–10161, 2021.

[115] Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. Cross-lingual name tagging and linking for 282 languages. In *Proc. of ACL*, pages 1946–1958, 2017.

[116] Shayne Longpre, Yi Lu, and Joachim Daiber. Mkqa: A linguistically diverse benchmark for multilingual open domain question answering. *arXiv preprint arXiv:2007.15207*, 2020.

[117] Cas W Coopmans, Helen De Hoop, Karthikeya Kaushik, Peter Hagoort, and Andrea E Martin. Hierarchy in language interpretation: evidence from behavioural experiments and computational modelling. *Language, Cognition and Neuroscience*, pages 1–20, 2021.

[118] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proc. of ACL*, pages 1715–1725, 2016.

[119] Kemal Kurniawan, Lea Frermann, Philip Schulz, and Trevor Cohn. Ppt: Parsimonious parser transfer for unsupervised cross-lingual adaptation. In *Proc. of EACL*, pages 2907–2918, 2021.

[120] Chenchen Ding, Ye Kyaw Thu, Masao Utiyama, and Eiichiro Sumita. Word segmentation for Burmese (Myanmar). *ACM Trans. Asian Low-Resour. Lang. Info. Process.*, 15(4):22, 2016.

[121] PAN Localization Cambodia (PLC) of IDRC. Research report on Khmer automatic POS tagging, 2008.

[122] PAN Localization Cambodia of IDRC. Part-of-speech template, 2007.

[123] Pakrigna Long and Veera Boonjing. Longest matching and rule-based techniques for Khmer word segmentation. In *Proc. of KST*, pages 80–83, 2018.

[124] Taku Kudo and Yuji Matsumoto. Chunking with support vector machines. In *Proc. of NAACL*, 2001.

[125] Benjamin Marie, Hour Kaing, Aye Myat Mon, Chenchen Ding, Atsushi Fujita, Masao Utiyama, and Eiichiro Sumita. Supervised and unsupervised machine translation for Myanmar-English and Khmer-English. In *Proc. of WAT*, pages 68–75, 2019.

[126] Rina Buoy and Sokchea Kor. Khmer word segmentation using BiLSTM networks. `https://github.com/rinabuoy/KhmerNLP`, 2020.

[127] Greg Durrett, Adam Pauls, and Dan Klein. Syntactic transfer using a bilingual lexicon. In *Proc. of EMNLP-CoNLL*, pages 1–11, 2012.

[128] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proc. of ACL*, pages 789–798, 2018.

[129] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proc. of AAAI*, pages 5012–5019, 2018.

[130] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proc. of EMNLP*, pages 2289–2294, 2016.

[131] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proc. of NAACL-HLT*, pages 1006–1011, 2015.

[132] Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proc. of NAACL-HLT*, 2003.

[133] Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. Cross-lingual bert transformation for zero-shot dependency parsing. In *Proc. of EMNLP*, pages 5721–5727, 2019.

[134] Dan Klein and Christopher D Manning. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proc. of ACL*, pages 478–485, 2004.

[135] Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. Learning to compose words into sentences with reinforcement learning. In *Proc. of ICLR*, 2017.

[136] Anders Søgaard. Data point selection for cross-language adaptation of dependency parsers. In *Proc. of ACL*, pages 682–686, 2011.

[137] Ayan Das, Agnivo Saha, and Sudeshna Sarkar. Cross-lingual transfer parser from hindi to bengali using delexicalization and chunking. In *Proc. of ICON*, pages 99–108, 2016.

[138] Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proc. of NAACL-HLT*, 2012.

[139] Hai Zhao, Yan Song, Chunyu Kit, and Guodong Zhou. Cross language dependency parsing using a bilingual lexicon. In *Proc. of AFNLP*, pages 55–63, 2009.

[140] Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. Zero-resource dependency parsing: Boosting delexicalized cross-lingual transfer with linguistic knowledge. In *Proc. of COLING*, page 119130, 2016.

[141] Ayan Das and Sudeshna Sarkar. Improving cross-lingual model transfer by chunking. *arXiv preprint arXiv:2002.12097*, 2020.

[142] Fei Xia and Martha Palmer. Converting dependency structures to phrase structures. In *Proc. of HLT*, 2001.

[143] Timothy Dozat and Christopher D Manning. Deep biaffine attention for neural dependency parsing. In *Proc. of ICLR*, 2017.

[144] Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. Cross-lingual transfer learning for pos tagging without cross-lingual resources. In *Proc. of EMNLP*, pages 2832–2838, 2017.

[145] Shijie Wu and Mark Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proc. of EMNLP-IJCNLP)*, pages 833–844, 2019.

[146] Yu Zhang, Houquan Zhou, and Zhenghua Li. Fast and accurate neural crf constituency parsing. In *Proc. of IJCAI*, pages 4046–4053, 2020.

[147] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *NeurIPS*, 2019.

[148] Eugene Charniak, Sharon Goldwater, and Mark Johnson. Edge-based best-first chart parsing. In *Sixth Workshop on Very Large Corpora*, 1998.

[149] Xinying Song, Shilin Ding, and Chin-Yew Lin. Better binarization for the cky parsing. In *Proc. of EMNLP*, pages 167–176, 2008.

[150] Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A Smith. What do recurrent neural network grammars learn about syntax? In *Proc. of EACL*, pages 1249–1258, 2017.

[151] Yuyu Zhang and Le Song. Language modeling with shared grammar. In *Proc. of ACL*, pages 4442–4453, 2019.

[152] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? In *Proc. of ACL*, pages 4996–5001, 2019.

Publication List

Journal Papers (peer-review)

1. **Hour Kaing**, Chenchen Ding, Masao Utiyama, Eiichiro Sumita, Sethserey Sam, Sopheap Seng, Katsuhito Sudoh, and Satoshi Nakamura. "Towards Tokenization and Part-of-Speech Tagging for Khmer: Data and Discussion." Transactions on Asian and Low-Resource Language Information Processing Vol. 20, No. 6, pp. 2375-4699, November 2021. Related to Chapter 3.

2. **Hour Kaing**, Chenchen Ding, Masao Utiyama, Eiichiro Sumita, Katsuhito Sudoh, and Satoshi Nakamura. "Constituency Parsing by Cross-Lingual Delexicalization." IEEE Access, Vol. 9, pp. 141571–141578, October 2021. Related to Chapter 4.

Conference Papers (peer-review)

1. **Hour Kaing**, Chenchen Ding, Katsuhito Sudoh, Masao Utiyama, Eiichiro Sumita, and Satoshi Nakamura. “Multi-Source Cross-Lingual Constituency Parsing.” The 18th International Conference on Natural Language Processing, 2021. Related to Chapter 5. (To Appear)

Other Papers

1. Benjamin Marie, **Hour Kaing**, Aye Myat Mon, Chenchen Ding, Atsushi Fujita, Masao Utiyama, and Eiichiro Sumita. "Supervised and unsupervised machine translation for Myanmar-English and Khmer-English." The 6th Workshop on Asian Translation, pp. 68–75. November 2019.