Auxiliary Lexicon Word Prediction for Cross-Domain Word Segmentation

Shohei Higashiyama^{†,††}, Masao Utiyama[†], Yuji Matsumoto^{†††},
Taro Watanabe^{††} and Eiichiro Sumita[†]

Recent work has explored various neural network-based methods for word segmentation and has achieved substantial progress mainly in in-domain scenarios. There remains, however, a problem of performance degradation on target domains for which labeled data is not available. A key issue in overcoming the problem is how to use linguistic resources in target domains, such as unlabeled data and lexicons, which can be collected or constructed more easily than fully-labeled data. In this work, we propose a novel method using unlabeled data and lexicons for cross-domain word segmentation. We introduce an auxiliary prediction task, Lexicon Word Prediction, into a character-based segmenter to identify occurrences of lexical entries in unlabeled sentences. The experiments demonstrate that the proposed method achieves accurate segmentation for various Japanese and Chinese domains.

Key Words: *Word Segmentation, Neural Networks, Domain Adaptation*

1 Introduction

Word segmentation is a fundamental step for many NLP applications in Asian languages such as Japanese and Chinese, in which there are no explicit word delimiters. In recent years, neural network models have been widely applied to word segmentation, especially for Chinese, to reduce the burden of manual feature engineering. These models are categorized as character-based or word-based. Character-based models (Zheng, Chen, and Xu 2013; Mansur, Pei, and Chang 2013; Pei, Ge, and Chang 2014; Chen, Qiu, Zhu, and Huang 2015a) predict segmentation label sequences by treating the task as character-level sequence labeling, while typical word-based models (Zhang, Zhang, and Fu 2016; Cai and Zhao 2016; Cai, Zhao, Zhang, Xin, Wu, and Huang 2017; Yang, Zhang, and Dong 2017) directly segment a character sequence into words by sequentially evaluating partial segmentation hypotheses. These neural models demonstrated large performance improvements for in-domain word segmentation. However, these models are

[†] National Institute of Information and Communications Technology

^{††} Nara Institute of Science and Technology

^{†††} RIKEN Center for Advanced Intelligence Project

based on supervised learning and require a large amount of manually labeled data to obtain satisfactory performance. Therefore, the main challenge in word segmentation is to achieve robust performance for out-of-domain texts.

Supervised and semi-supervised methods using various linguistic resources in target domains have also been explored for cross-domain word segmentation. Lexicons and unlabeled data are especially good complementary resources, both of which can be collected or constructed more easily than fully-labeled data. A lexicon feature (Neubig, Nakata, and Mori 2011; Zhang, Zhang, Che, and Liu 2014; Wu, He, Zhong, Zhou, and Yuan 2014) is a well-known technique that uses occurrence information of lexical entries in a given sentence. However, models based on lexical features may not sufficiently adapt to target domains since they cannot learn the proper relationship between feature values and segmentation labels for unlabeled target sentences. Another technique, called distant supervision (Mintz, Bills, Snow, and Jurafsky 2009), uses pseudo labeled data generated from unlabeled data and a lexicon. Liu, Zhang, Che, Liu, and Wu (2014) and Zhao, Zhang, Wang, and Liu (2018) augmented labeled data with pseudo partially-labeled data generated by matching lexical entries with unlabeled sentences. However, pseudo labeled data can be noisy because a heuristic matching method, e.g., the longest matching, may not correctly resolve the ambiguities that different lexical entries can match within overlapping spans in a sentence.

In this work, we propose a novel word segmentation method using unlabeled data and lexicons. We introduce an auxiliary task, which we call Lexicon Word Prediction (LWP), into a character-based segmenter to predict whether each character in a sentence corresponds to a particular position of a word retrieved from a lexicon. With the help of the auxiliary task, a model learns word indicators from unlabeled (source and) target sentences, together with segmentation label information from source labeled sentences. This method can naturally handle conflicts of lexicon matching by introducing multiple LWP tasks to predict different positions; a character in a sentence can be the beginning, middle, or end of different words simultaneously.

The contributions of this research are as follows:

- Introduction of a novel word segmentation method to learn explicit signals of word occurrence with surrounding contexts from unlabeled sentences.
- A demonstration that the method improves performance for various target domains, while preventing performance degradation for source and other domains.
- Achieving better or competitive performance on Japanese and Chinese datasets, compared with existing methods for cross-domain word segmentation.

2 Character-Based Word Segmentation

2.1 Task Definition

Word segmentation can be seen as a character-level sequence labeling task. Labeled data $\mathcal{D}_l = \{(x, t)\}$ is a set of pairs of a sentence $x = x_{1:n} = (x_1, \dots, x_n)$ and its gold label sequence $t = t_{1:n} = (t_1, \dots, t_n)$, where a character x_i is in a character vocabulary $\mathcal{V}_c$ and a label t_i is in a tag set $\mathcal{T}$. Given a sentence x , a segmentation model is required to predict a label sequence $y = y_{1:n} = (y_1, \dots, y_n) \in \mathcal{T}^n$. We employ tag set $\mathcal{T} = \{B, I, E, S\}$, where B, I and E represent the beginning, inside and end of a multi-character word, and S represents a single character word (Xue 2003).

2.2 BiLSTM Model for Word Segmentation

As a baseline, we use a BiLSTM model (Huang, Xu, and Yu 2015; Chen, Qiu, Zhu, Liu, and Huang 2015b) that is a standard architecture for character-based word segmentation. The model consists of a character embedding layer, BiLSTM layers, and an inference layer.

Character Embedding Layer For each character x_i in a given sentence x , a character embedding $\mathbf{e}_i$ of a d_c -dimensional vector is retrieved from a character embedding matrix $E_c \in \mathbb{R}^{d_c \times |\mathcal{V}_c|}$. We randomly initialized character embeddings, since pre-trained character embeddings did not improve performance in our preliminary experiments.

BiLSTM Layers Embedding vectors $\mathbf{e}_{1:n} = (\mathbf{e}_1, \dots, \mathbf{e}_n)$ for a sentence are fed into an N -layer recurrent neural network (RNN) to derive contextualized representations $\mathbf{h}_{1:n}^{(l)} = (\mathbf{h}_1^{(l)}, \dots, \mathbf{h}_n^{(l)})$ ($l \in \{1, \dots, N\}$). We adopt a bidirectional variant of a long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) network, which addresses the issue of learning long-term dependencies and the gradient vanishing problem. Bidirectional LSTM (BiLSTM) consists of forward and backward LSTMs. An l -th BiLSTM layer concatenates a forward hidden vector $\vec{\mathbf{h}}_i^{(l)} \in \mathbb{R}^{d_r}$ and a backward hidden vector $\overleftarrow{\mathbf{h}}_i^{(l)} \in \mathbb{R}^{d_r}$, which are calculated by forward LSTM (LSTM_f) and backward LSTM (LSTM_b). The outputs is a hidden vector $\mathbf{h}_i^{(l)} \in \mathbb{R}^{2d_r}$ for each time step i :

$$\begin{aligned}\vec{\mathbf{h}}_i^{(l)} &= \text{LSTM}_f(\mathbf{h}_i^{(l-1)}, \vec{\mathbf{h}}_{i-1}^{(l)}) , \\ \overleftarrow{\mathbf{h}}_i^{(l)} &= \text{LSTM}_b(\mathbf{h}_i^{(l-1)}, \overleftarrow{\mathbf{h}}_{i+1}^{(l)}) , \\ \mathbf{h}_i^{(l)} &= \vec{\mathbf{h}}_i^{(l)} \oplus \overleftarrow{\mathbf{h}}_i^{(l)} ,\end{aligned}$$

where $\oplus$ denotes a concatenation operation, $\mathbf{h}_i^{(0)} = \mathbf{e}_i$, and d_r is a hyperparameter that corresponds to the number of rows in the LSTM parameter matrices. Note that the number of columns

in the parameter matrices is equal to the dimension of an input vector $\mathbf{h}_i^{(l-1)}$.

Inference Layer An output vector $\mathbf{h}_i = \mathbf{h}_i^{(N)}$ of the BiLSTM for character x_i is mapped into a $|\mathcal{T}|$ -dimensional vector representing scores of segmentation labels via an affine layer:

$$\mathbf{s}_i = W_s \mathbf{h}_i + \mathbf{b}_s ,$$

where $W_s \in \mathbb{R}^{|\mathcal{T}| \times 2d_r}$ and $\mathbf{b}_s \in \mathbb{R}^{|\mathcal{T}|}$ are trainable parameters. The model outputs a predicted label sequence $y = y_{1:n}$ for a sentence x , where each label $y_i \in \mathcal{T}$ corresponds to the dimension of the score vector $\mathbf{s}_i$ with the largest value. Note that we did not adopt the CRF-based prediction since it did not saliently outperform softmax-based one in our preliminary experiments.

Training Objective During training, the parameters of the network are learned by minimizing the loss function L_{seg} for the segmentation task, which is defined as the cross entropy between gold and predicted label distributions:

$$L_{\text{seg}}(D_l) = - \sum_{(x,t) \in \mathcal{D}_l} \sum_i \mathbf{t}_i \log \mathbf{y}_i , \quad (1)$$

$$\mathbf{y}_i = \frac{\exp(\mathbf{s}_i)}{\sum_k \exp(s_{i,k})} ,$$

where $\mathbf{t}_i$ indicates the one-hot vector of the gold label t_i , $\mathbf{y}_i$ indicates predicted label distribution, and $s_{i,k}$ indicates the k -th element of the score vector $\mathbf{s}_i$.

3 Lexicon Word Prediction for Cross-Domain Word Segmentation

In addition to labeled data $\mathcal{D}_l$, we assume that unlabeled data $\mathcal{D}_u = \{x\}$ and a word lexicon $\mathcal{L} = \{w\}$ are available. A word w in a lexicon is a sequence of characters. The j -th character of a word w is denoted as w_j and the length of a word as $|w|$. Although the proposed auxiliary task is introduced to a BiLSTM model, it can be used with any neural architectures including convolutional neural networks (CNNs) (LeCun, Bottou, Bengio, and Haffner 1998; Collobert and Weston 2008) and self-attention networks (SANs) (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin 2017).

Lexicon Word Prediction An explicit lexicon feature is expected to capture a word occurrence in a context. However, labeled data in a new domain is necessary to learn the proper weights of the features for sentences in that domain. Instead, we introduce a novel auxiliary task to adapt to a new domain using unlabeled data and a lexicon. This method performs joint learning of the segmentation task based on labeled data and the auxiliary task based on unlabeled data.

Lexicon:

$$\mathcal{L} = \{ \text{“データ” (data), “データベース” (database), “ベース” (base), “を” (ACC), “作” (make), “作成” (create), “成” (consist), “する” (do)} \}$$

Sentence:

$$x = (\text{デ} \text{ー} \text{タ} \text{ベ} \text{ー} \text{ス} \text{を} \text{作} \text{成} \text{す} \text{る})$$

Auxiliary label sequences:

$$u^{(B)} = (1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0)$$

$$u^{(I)} = (0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)$$

$$u^{(E)} = (0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1)$$

$$u^{(S)} = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0)$$

Fig. 1 An example of auxiliary label sequences $u^{(B)}$, $u^{(I)}$, $u^{(E)}$, and $u^{(S)}$ of the B, I, E, and S positions for a lexicon $\mathcal{L}$ and a sentence x , which means “create a database”

The Lexicon Word Prediction (LWP) auxiliary task is defined as follows: a model predicts whether each character in a sentence corresponds to a particular position of a word in a lexicon. Formally, an auxiliary label sequence $u^{(B)} = u_{1:n}^{(B)} \in \{0, 1\}^n$ is generated for each (labeled or unlabeled) sentence $x = x_{1:n}$ by matching substrings of x with any words in a lexicon $\mathcal{L}$. Given a character position j and a word length k such that $j = 1 < k$, an auxiliary label $u_i^{(B)}$ for a character x_i , which indicates whether x_i corresponds to the beginning of a word w (B), is defined as follows:

$$u_i^{(B)} = \begin{cases} 1 & (\exists w \in \mathcal{L}, x_i = w_j \text{ and } |w| = k) , \\ 0 & (\text{otherwise}) . \end{cases} \quad (2)$$

Similarly, auxiliary label sequences $u^{(I)}$, $u^{(E)}$, and $u^{(S)} \in \{0, 1\}^n$ are generated to indicate the inside of a word (I), the end of a word (E), and a single character word (S). Each label $u_i^{(I)}$, $u_i^{(E)}$, and $u_i^{(S)}$ is similarly defined by letting $1 < j < k$, $j = k > 1$, and $j = k = 1$ in Eq. (2), respectively. Fig. 1 illustrates an example of auxiliary label sequences for a sentence and a lexicon.

The loss of the auxiliary task is defined for each position by the cross entropy based on auxiliary labels, similarly to the segmentation loss L_{seg} in Eq. (1). Then, the auxiliary loss L_{aux} is the sum of the losses for four LWP tasks w.r.t B, I, E and S positions:

$$L_{\text{aux}}(\mathcal{D}_l \cup \mathcal{D}_u) = - \sum_{(x,t) \in \mathcal{D}_l \cup \mathcal{D}_u} \sum_{p \in \{B, I, E, S\}} \sum_i \mathbf{u}_i^{(p)} \log \mathbf{y}_i^{(p)} ,$$

where $\mathbf{u}_i^{(p)}$ ($p \in \{B, I, E, S\}$) is the one-hot vector of the auxiliary label and $\mathbf{y}_i^{(p)}$ is the predicted

label distribution. Note that any lexical information except for auxiliary labels are not given for solving the LWP task.

Finally, the weighted sum of the loss functions of both tasks is minimized:

$$L_{\text{seg}}(\mathcal{D}_l) + \lambda L_{\text{aux}}(\mathcal{D}_l \cup \mathcal{D}_u) , \quad (3)$$

where λ is a hyperparameter to control the importance of the auxiliary task. Note that, as for labeled sentences, the model is trained not only on the segmentation task but also on the auxiliary task.

Task-Specific MLPs As additional components, we introduce multi-layer perceptrons (MLPs) to learn task-specific representations. Let d_m be a hyperparameter. A hidden vector $\mathbf{h}_i$ from the BiLSTM layers is transformed into task-specific vectors $\mathbf{m}_i^{(\text{seg})}, \mathbf{m}_i^{(\text{aux})} \in \mathbb{R}^{d_m}$ via different MLPs with one hidden layer for the main task (MLP_{seg}) and the auxiliary task (MLP_{aux}):

$$\begin{aligned} \mathbf{m}_i^{(\text{seg})} &= \text{MLP}_{\text{seg}}(\mathbf{h}_i) = g(U_s \mathbf{h}_i + \mathbf{v}_s) , \\ \mathbf{m}_i^{(\text{aux})} &= \text{MLP}_{\text{aux}}(\mathbf{h}_i) = g(U_a \mathbf{h}_i + \mathbf{v}_a) , \end{aligned}$$

where g indicates the ReLU activation function, and $U_s, U_a \in \mathbb{R}^{d_m \times 2d_r}$ and $\mathbf{v}_s, \mathbf{v}_a \in \mathbb{R}^{d_m}$ are trainable parameters. Then, the task-specific vector for each task is transformed into a score vector $\mathbf{s}_i^{(\text{seg})} \in \mathbb{R}^{|\mathcal{T}|}$ or $\mathbf{s}_i^{(\text{aux-}p)} \in \mathbb{R}^2$ for $p \in \{B, I, E, S\}$, respectively:

$$\begin{aligned} \mathbf{s}_i^{(\text{seg})} &= W_s \mathbf{m}_i^{(\text{seg})} + \mathbf{b}_s , \\ \mathbf{s}_i^{(\text{aux-}p)} &= W_{a,p} \mathbf{m}_i^{(\text{aux})} + \mathbf{b}_{a,p} , \end{aligned}$$

where $W_s \in \mathbb{R}^{|\mathcal{T}| \times d_m}$, $W_{a,p} \in \mathbb{R}^{2 \times d_m}$, $\mathbf{b}_s \in \mathbb{R}^{|\mathcal{T}|}$, and $\mathbf{b}_{a,p} \in \mathbb{R}^2$ are trainable parameters. The model outputs predicted label distributions for the segmentation and auxiliary tasks similarly to the baseline method described in §2.2, and the loss is calculated by Eq. (3).

4 Experiments

4.1 Language Resources

Datasets For Japanese experiments, we used Japanese Dependency Corpus¹ (JDC) (Mori, Ogura, and Sasada 2014). The corpus consists of the core data of BCCWJ^{2,3} (Maekawa,

¹ <http://www.ar.media.kyoto-u.ac.jp/data/word-dep/JDC-2015-06.tar.gz>

² <https://pj.ninjal.ac.jp/corpus.center/bccwj/en/>

³ BCCWJ is a Japanese balanced corpus including books, news articles, magazines, and blogs etc.

Table 1 Dataset statistics

	Japanese				Chinese		Chinese					
	Source	Target			Source	Target	Source	Target				
	GEN	JNL	JPT	RCP	CTB	C-ZX	PKU	P-ZX	FR	DL	DM	CPT
Train	30K	—	—	—	18K	—	17K	—	—	—	—	—
Dev	0.5K	—	—	—	0.4K	(0.8K)	2.0K	(0.3K)	—	—	—	—
Test	2.0K	0.3K	2.0K	0.7K	0.3K	1.4K	2.0K	0.7K	1.0K	1.0K	1.0K	1.0K
Unlabeled	480K	480K	480K	480K	480K	27K	480K	27K	130K	40K	46K	
Lexicon	570K	13K	134K	21K	390K	0.5K	390K	0.5K	0.7K	0.5K	0	0

Yamazaki, Ogiso, Maruyama, Ogura, Kashino, Koiso, Yamaguchi, Tanaka, and Den 2014), economic newspaper articles, dictionary example sentences, computer science journal abstracts (JNL), patent specifications (JPT), and recipes (RCP). We used sentences in the former three domains as the source domain data, which we called GEN, and randomly selected 500 and 2000 sentences for the development set and the in-domain test set. We used JNL, JPT,⁴ and RCP domain data for cross-domain evaluation.

For Chinese experiments, we used two source domain data: Chinese Treebank (CTB) 5.0⁵ and the SIGHAN Bakeoff 2005 (Emerson 2005) PKU data. As evaluation data for CTB, we used an internet novel dataset Zhuxian (C-ZX) (Zhang et al. 2014). As evaluation data for PKU, we used three internet novel datasets (Qiu and Zhang 2015), Zhuxian (P-ZX), FanrenXiuxianZhuan (FR), and DouluoDalu (DL) along with two science and technology datasets (Qiu, Shi, and Wang 2015) dermatology (DM) and patent (CPT). These combinations of source and target domain data were adopted for comparison with previous work. C-ZX and P-ZX were from the same data source but had the different data splits introduced by previous work (Zhang et al. 2014; Qiu and Zhang 2015). We followed the training/development/test split of CTB by Zhang and Clark (2011) and the official training/test split of PKU. We used randomly sampled 90% of sentences of the PKU training data as the training set and the remaining sentences as the development set. We normalized texts in the Chinese datasets by converting single-byte characters to double-byte ones as preprocessing.

Table 1 shows the dataset statistics. Values in the train, dev, test, and unlabeled rows indicate the numbers of sentences, and values in the lexicon row indicate the number of entries. Note that the development sets of C-ZX and P-ZX were not used in our experiments.

Unlabeled Data For in-domain experiments, we used unlabeled data in source domains: the

⁴ We converted HTML entities (e.g., “>”) contained in JPT to corresponding characters (e.g., “>”).

⁵ <https://catalog.ldc.upenn.edu/LDC2005T01>

non-core data of BCCWJ Version 1.1⁶ for GEN and Chinese Gigaword Fifth Edition⁷ for CTB and PKU.

As unlabeled data for cross-domain experiments, we used Japanese computer science paper abstracts published on IPSJ Digital Library⁸ for JNL, NTCIR-8 PATMT Test Collection⁹ for JPT, the “steps” portion of the Cookpad Dataset¹⁰ for RCP. We used the unlabeled data provided by Ye, Li, Zhang, Qiu, and Sun (2019)¹¹ for the target domains of the Chinese datasets. We used the same unlabeled data from Zhuxian for C-ZX and P-ZX. The unlabeled data of the novel domains included raw test sentences.

Lexicon We used UniDic¹² (Den, Ogiso, Ogura, Yamada, Minematsu, Uchimoto, and Koiso 2007) and Jieba dictionary¹³ as source lexicons for the Japanese and Chinese datasets, respectively. In addition, we constructed target lexicons from keywords in Japanese computer science papers for JNL, from JST thesaurus¹⁴ for JPT, from the “ingredients” portion of the Cookpad Dataset for RCP, and from the articles on the novels in Baidu Baike and Chinese Wikipedia for the Chinese novel domains. We also used the Zhuxian name lexicon¹⁵ for C-ZX and P-ZX.

The preprocessing steps of lexicon construction were as follows. For the JNL lexicon, we split each keyword in computer science papers by predefined expressions and used separated strings.¹⁶ For the JPT lexicon, we used both the original entries in JST thesaurus and the auto-segmented results of the entries. For the RCP lexicon, we split each ingredient description by punctuation and coordinate conjunctions¹⁷ and adopted strings occurring at least 10 times. For each novel domain lexicon, we collected entity names by extracting strings surrounded by particular XML tags from the corresponding encyclopedia pages. While these semi-automatically constructed (or extended) lexicons often contain multi-words or phrases rather than words, we avoided the manual checking cost by using them as they were.

⁶ We restored the provided auto-segmented data to the original raw sentences and used them as unlabeled data.

⁷ <https://catalog.ldc.upenn.edu/LDC2011T13>

⁸ https://www.ipsj.or.jp/e-library/digital_library.html (in Japanese)

⁹ <http://research.nii.ac.jp/ntcir/permission/ntcir-8/perm-en-PATMT.html>

¹⁰ <https://www.nii.ac.jp/dsc/idr/cookpad/> (in Japanese)

¹¹ <https://github.com/vatile/CWS-NAACL2019>

¹² https://unidic.ninjal.ac.jp/unidic_archive/cwj/2.1.2/unidic-mecab-2.1.2_src.zip

¹³ <https://github.com/foxsjy/jieba/blob/master/jieba/dict.txt>

¹⁴ <https://dbarchive.biosciencedbc.jp/en/mecab/data-1.html>

¹⁵ https://github.com/egrcc/Cross-Domain-CWS/blob/master/dataset/preprocess_data/zx/zx_dict.txt

¹⁶ We investigated frequent functional expressions that occurred in between noun phrases in keywords and used “および”/“及び” (and), “とその”, (and that) “における” (on), “による” (by), “への” (to), “からの” (from), “のための” (for), “に向けた” (toward), “する” (do), and single-character particles, such as “が” (nominative case) and “と” (coordinate conjunction), as the predefined expressions. E.g., an original keyword “機械学習と自然言語処理” (machine learning and natural language processing) is split into “機械学習” and “自然言語処理.”

¹⁷ For example, an original description “牛肉または豚肉” (beef or pork) is split into “牛肉” and “豚肉.”

We merged a source lexicon and corresponding target lexicon(s) into a single lexicon for each target domain and used it to train each domain-specific model. Since there were no target lexicons for DM and CPT, a single model for these domains was trained using the source lexicon and the merged target unlabeled data in two domains.

4.2 Settings

Baseline Methods We used the following three baselines.

- A naïve baseline (BL): A BiLSTM model, which is described in §2.2, trained from source labeled data.
- A self-training baseline (ST): A BiLSTM model trained from labeled data in source domain and auto-segmented data in each target domain. BL was applied to target unlabeled data to obtain auto-segmented data.
- A lexicon feature baseline (LF): A BiLSTM model enhanced with lexicon features and trained from source labeled data. Binary lexicon features were defined that indicated whether a character corresponded to a particular position (immediate left, immediate right, beginning, middle, or end) of any lexical word of length k ($2 \leq k \leq 3$, $4 \leq k \leq 5$, or $6 \leq k \leq 10$). A fixed-sized¹⁸ vector $\mathbf{l}_i$ was constructed for a character x_i and used $\mathbf{e}'_i = \mathbf{e}_i \oplus \mathbf{l}_i$ as input to BiLSTM layers, instead of the character embedding $\mathbf{e}_i$. Differing from the previous work (Zhang, Liu, and Fu 2018) that extended a standard LSTM architecture to incorporate lexicon features, these features were used in the above simple manner.

Training Setting Suppose there were n_l labeled sentences and n_u unlabeled sentences. To keep training time manageable for a large amount of unlabeled data, for each training epoch, we used $2n_l$ training sentences consisting of all labeled sentences and randomly sampled n_l unlabeled sentences. In each iteration, we alternately made a mini-batch consisting only of labeled or unlabeled sentences. Only L_{aux} was calculated in Eq. (3) for mini-batches consisting of unlabeled sentences. In this way, we trained a domain-specific model for each target domain. We adopted a similar strategy for training the ST baseline using auto-segmented sentences instead of raw unlabeled sentences.

Table 2 gives the hyperparameters for the baseline and proposed methods. We used lexical words whose length was less or equal to 6 when generating auxiliary labels, because there were many cases in which longer lexical entries in target lexicons were not single words.¹⁹ We applied

¹⁸ The dimension was $15 = 5 \text{ positions} \times 3 \text{ length groups}$.

¹⁹ For example, the target lexicon for RCP has multi-word entries such as “トマトジュース” (tomato juice), “しょうゆ大さじ1” (a tablespoon of soy sauce), and “中華スープの素” (Chinese soup mix).

Table 2 Hyperparameter values for the baseline and proposed methods

Method	Hyperparameter	Value
Baseline/Proposed method	Character embedding size (d_c)	300
	Number of BiLSTM layers	2
	Number of BiLSTM hidden units (d_r)	600
	Mini-batch size	100
	Initial learning rate	1.0
	Learning rate decay rate	0.9
	Gradient clipping threshold	5.0
	Recurrent dropout rate	0.4
Proposed method	Number of MLP hidden units (d_m, d_a)	300
	Weight for auxiliary loss (λ)	0.25
	Minimum word length	1
	Maximum word length	6

Table 3 Performance on source domain test sets

Method	Resource	JDC	CTB	PKU
BL	–	98.0	96.9	94.5
ST	$\mathcal{U}_s$	98.1*	96.8	94.6*
LF	$\mathcal{L}_s$	98.5*	97.0*	95.6*
LWP-S	$\mathcal{U}_s, \mathcal{L}_s$	98.4* [†]	96.7 [†]	95.3* ^{†‡}

The resource column lists resources used by each method: source unlabeled data $\mathcal{U}_s$ and source lexicon $\mathcal{L}_s$.

dropout (Zaremba, Sutskever, and Vinyals 2015) to non-recurrent connections of recurrent layers. We used a mini-batch stochastic gradient descent to optimize parameters and decayed the learning rate with a fixed decay rate every epoch after the first five epochs. We trained models for up to 20 epochs and used early stopping based on the F1 score on the development set.

4.3 Main Results

4.3.1 In-domain results

We evaluated the baseline methods and the proposed method with source domain resources (LWP-S) in the in-domain setting, expecting our auxiliary task to encode word occurrence information and to work similarly to a lexicon feature in source domains. Table 3 shows the mean F1 score of three runs for each method and each dataset. We conducted McNemar’s tests (Gillick and Cox 1989) on the differences between word-level predictions (true positive or false negative) of two systems for gold words. The symbols $\star$, $\dagger$, and $\ddagger$ in Table 3 indicate statistical significance at the 0.001 level over the BL, ST, and LF, respectively. The symbol $\ddagger$ indicates that the

Table 4 Accuracy of LWP-S on auxiliary label classification on the test sets

Position	JDC	CTB	PKU
B	99.4 (43.5)	98.5 (43.2)	97.9 (41.3)
I	99.4 (11.1)	97.6 (14.2)	97.1 (12.0)
E	99.3 (45.2)	98.5 (44.4)	97.9 (43.0)
S	100.0 (99.2)	100.0 (99.2)	100.0 (99.8)
Total	99.5 (49.8)	98.7 (50.3)	98.2 (49.1)

Values in “()” indicate the percentage of positive labels ($u_i^{(p)} = 1$) in all labels for each position $p \in \{B, I, E, S\}$.

performance is significantly lower than that of the LF.

The improvements of the ST and the LWP-S over the BL were significant on JDC and PKU, and those of the LF over the BL were significant on three datasets. The performance differences between the ST and the LWP-S were significant on three domains,²⁰ and that between the LF and the LWP-S was significant only on PKU. Compared to the LF, the proposed method achieved similar but slightly lower segmentation performance; the LF has the advantage that it accesses information on all words in a lexicon, while the proposed method only uses information encoded in the model via pseudo labels during training.

Table 4 shows the mean accuracy of three runs of the LWP-S on auxiliary label classification on the test sets. Our method yielded at least 97% accuracy for each position while the overall performance was biased toward the easiest S position. These results supported the expectation; our method successfully learns word occurrence information and exploits it for segmentation decisions.

4.3.2 Cross-domain results

In the cross-domain setting, we evaluated the baseline methods and the proposed method with source domain resources (LWP-S) or target domain resources (LWP-T) on the target domain test sets. Table 5 shows the mean F1 score of three runs for each method and each dataset. The symbols $\star$, $\dagger$, and $\ddagger$ in Table 5 indicate statistical significance at the 0.001 level over the BL, ST, and LF, respectively, according to the McNeamer’s tests similar to the in-domain experiments. The symbol $\star$ indicate that the performance is significantly lower than that of BL.

The ST showed limited improvements (+0.2 points over the BL on average). The LF showed

²⁰ Our recall-oriented significance tests showed that the improvement of the LWP-S (recall of 97.6) over the ST (97.3) on CTB was significant, although the ST performed better in terms of F1 score as in Table 3.

Table 5 Performance on target domain test sets

Method	Resource	JNL	JPT	RCP	C-ZX	P-ZX	FR	DL	DM	CPT
BL	–	97.2	95.0	94.3	86.3	82.4	84.9	87.8	80.5	86.6
ST	$\mathcal{U}_t$	97.4*	94.8*	94.8*	86.8*	82.9*	86.1*	87.9	80.1	86.1
LF	$\mathcal{L}_s \cup \mathcal{L}_t$	97.5*	96.3*	94.4*	90.4*	87.6*	86.0*	89.5*	82.7*	88.3*
LWP-S	$\mathcal{U}_s, \mathcal{L}_s$	97.8	97.0	95.3	88.5	83.8	86.8	88.7	82.2	88.2
LWP-T	$\mathcal{U}_t, \mathcal{L}_s \cup \mathcal{L}_t$	98.2*†‡	97.6*†‡	95.4*†‡	91.7*†‡	89.7*†‡	87.4*†‡	90.7*†‡	83.8*†‡	89.4*†‡
LWP-O	$\mathcal{U}_t, \mathcal{L}_s \cup \mathcal{V}_{w:\text{test}}$	98.4	98.5	96.2	93.3	92.1	93.4	94.1	90.0	93.3

The resource column lists resources used by each method: target unlabeled data $\mathcal{U}_t$, target lexicon $\mathcal{L}_t$, and oracle lexicon $\mathcal{V}_{w:\text{test}}$ (i.e., the set of gold words in the test set). Both LF and LWP-T used only the source lexicon on DM and CPT (i.e., $\mathcal{L}_t = \emptyset$ for these domains).

Table 6 OOV rate of a test set for a vocabulary

OOV rate for vocabulary	JNL	JPT	RCP	C-ZX	P-ZX	FR	DL	DM	CPT
$\mathcal{V}_0 = \mathcal{V}_{w:\text{train}}$	5.58	9.48	6.87	15.44	16.18	13.94	11.03	22.16	15.18
$\mathcal{V}_s = \mathcal{V}_{w:\text{train}} \cup \mathcal{L}_s$	1.32	2.14	1.04	5.04	7.63	7.68	6.59	9.95	7.19
$\mathcal{V}_t = \mathcal{V}_{w:\text{train}} \cup \mathcal{L}_s \cup \mathcal{L}_t$	0.66	1.73	0.95	2.06	5.01	5.64	3.70	9.95	7.19
$\Delta(\mathcal{V}_t, \mathcal{V}_0)$	4.92	7.75	5.92	13.38	11.17	8.30	7.33	12.21	7.99

$\mathcal{V}_{w:\text{train}}$ indicate the set of gold words in the corresponding training set. Δ indicates the difference of the rates between $\mathcal{V}_t$ and $\mathcal{V}_0$.

a more clearly improved performance (+2.0 points over the BL). The proposed method, LWP-T, achieved larger improvements (+3.2 points over the BL) than the other enhanced baselines on all domains. These results validate our auxiliary task that enables it to learn word indicators in target contexts. The performance of the ST, LF, and LWP-T was significantly better than that of the BL on 6, 8, and 9 out of 9 datasets, respectively.²¹ Moreover, the performance of the LWP-T was significantly better than that of the ST and the LF on all domains. Note that the performance of the LWP-S was also significantly better than that of the BL and the ST on all domains and than that of the LF on the three Japanese domains.

Table 6 shows the out-of-vocabulary (OOV) rate of a test set for a vocabulary $\mathcal{V}$. OOV rate indicates the percentage of OOV word tokens, which are word tokens not contained in $\mathcal{V}$, in all word tokens in the test set.²²

²¹ Our significance tests showed that the improvement of the ST (recall of 94.9) over the BL (94.8) on JPT, and the degradation of the LF (recall of 94.5) over the BL (95.0) on RCP were significant, although the BL performed better in the former case and the LF performed better in the latter case in terms of F1 score as in Table 5.

²² For example, the OOV rate of a test set consisting of 6 words, “A,” “A,” “B,” “C,” “D” and “E,” for a vocabulary $\mathcal{V} = \{A, B, C\}$ is $2/6 = 33.3\%$.

Table 7 Comparison with state-of-the-art methods on test sets

Method	UL	Lex	JNL	JPT	RCP	C-ZX	P-ZX	FR	DL	DM	CPT
Ours	✓	✓	98.2	97.6	95.4	91.7	89.7	87.4	90.7	83.8	89.4
Neubig et al. (2011) [◦]		✓	97.8	97.2	95.5	—	—	—	—	—	—
Kitagawa and Komachi (2018)			97.6	93.1	94.0	—	—	—	—	—	—
Higashiyama et al. (2019)	✓		98.1	96.7	95.2	—	—	—	—	—	—
Liu et al. (2014) [◦]	✓	✓	—	—	—	90.6	—	—	—	—	—
Zhou et al. (2017)	✓		—	—	—	90.1	—	—	—	—	—
Zhao et al. (2018)	✓	✓	—	—	—	92.9	—	—	—	—	—
Zhang and Clark (2010) [◦]			—	—	—	—	86.8	85.9	90.5	77.9	84.6
Ye et al. (2019)	✓		—	—	—	—	89.6	89.6	93.5	82.2	85.1
Gan and Zhang (2019)	✓		—	—	—	—	90.5	91.1	93.0	—	—

“UL” and “lex” indicates whether a method uses additional unlabeled data and lexicons, respectively. Non-neural methods are marked with “[◦]”.

The results led to the following findings. First, there was a tendency for the proposed method to yield larger performance improvements on domains in which the OOV rate largely decreased by adding a lexicon. We observed improvements of more than 3 F1 points on C-ZX, P-ZX, and DM (OOV rate reduction of more than 11 points), those of more than 2.5 F1 points on JPT, FR, DL, and CPT (OOV rate reduction of more than 7 points), and those of about 1 F1 point on JNL and RCP (OOV rate reduction of about 5 or 6 points). Second, the proposed method is not sensitive to the size of unlabeled data; the smaller size of unlabeled data used for Chinese domains as in Table 1 did not stop the large improvements on these domains, probably due to the OOV rate reduction as above. This also suggests that a reasonable size of unlabeled data covers frequent words in a target domain. Third, the proposed method using only a source lexicon was effective when combining source unlabeled data (LWP-S on all domains) or target unlabeled data (LWP-T on DM and CPT). This concludes that the proposed method is applicable to broad domains including low resource domains where off-the-shelf lexicons are not available.

For reference, we evaluated the proposed method using the oracle lexicon (LWP-O), i.e., the set of gold words in the test set, instead of the original target lexicons. The results are shown in the last column of Table 5. The higher performance of the LWP-O demonstrates that the proposed method can achieve further improvements using a higher-coverage lexicon.

4.3.3 Comparison with state-of-the-art methods

Table 7 shows results of state-of-the-art methods. The results of the three methods in the second block are from our run on their implementations, and those of the methods in the third block are cited from their papers (the result of Zhang and Clark (2010) is cited from Ye et al.

Fig. 2 Performance of LWP-T for each λ value on the JPT and C-ZX test sets

(2019)). We cited the results of Gan and Zhang (2019)’s method that did not rely on POS information for comparison of methods based on unlabeled data and/or word lexicons.

Our method achieved better performance than existing methods on some domains (JNL, JPT, DM, and CPT) and competitive performance on the other domains, while direct comparison was difficult since each method relied on different unlabeled data or lexicons. Note that our method was on a par with Zhao et al. (2018)’s method that incorporated partially-labeled target sentences (F1 of 91.6 on C-ZX) but their method obtained further gains (+1.3 points as in Table 7) by integrating a character-level language model. Similarly, Gan and Zhang (2019) showed improvements by introducing BERT (Devlin, Chang, Lee, and Toutanova 2019) character embeddings. Our method may also obtain benefits from combining language modeling-like information learned from a huge amount of data.

4.4 Analysis

4.4.1 Influence of weight for auxiliary loss

We investigated the influence of the hyperparameter λ to control the importance of the LWP task in Eq. (3). Fig. 2 shows F1 scores of single runs of the LWP-T with different λ values (0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1, and 2) on the JPT and C-ZX test sets. According to the increase of the value of λ , the performance of the proposed method improved from the baseline performance with $\lambda = 0$. However, it was saturated when λ was moderate (around 0.1) and gradually degraded for larger λ values due to over-emphasized loss values of the LWP task. This

tendency was consistent for both domains.

4.4.2 Performance of adapted models on various domains

We regard a domain of unlabeled data used for training by our method as an adaptation domain. We evaluated each adapted model (LWP-S or LWP-T) on other domains than the adaptation domain. Table 8 shows the mean F1 score of three runs for each dataset. The following was observed: (1) as expected, the adapted models performed the best on the adaptation domains (except CTB) compared with the models adapted to other domains; (2) the models adapted to any target domains performed better than the BL on the source domains (except CTB) and performed similarly to or better than the BL on the irrelevant domains, which were neither the source nor the target domains. These results show that the proposed method can adapt to a target domain, while preventing performance degradation on source and other domains.

4.4.3 Performance on unknown words

We examined performance of the proposed method on unknown words, i.e., words not oc-

Table 8 Performance of models adapted to adaptation domains (“adapt”) on the test sets of evaluation domains (“eval”)

(a) JDC						(b) CTB			
Method	Adapt\Eval	GEN	JNL	JPT	RCP	Method	Adapt\Eval	CTB	C-ZX
BL	GEN	98.0	97.2	95.0	94.3	BL	CTB	96.9	86.3
LWP-S	GEN	98.4	97.8	97.0	95.3	LWP-S	CTB	96.7	88.4
LWP-T	JNL	98.1	98.2	97.4	94.8	LWP-T	C-ZX	96.5	91.7
	JPT	98.2	97.7	97.6	94.8				
	RCP	98.2	97.2	95.8	95.4				

(c) PKU							
Method	Adapt\Eval	PKU	P-ZX	FR	DL	DM	CPT
BL	PKU	94.5	82.4	84.9	87.8	80.5	86.6
LWP-S	PKU	95.3	83.8	86.8	88.7	82.2	88.2
LWP-T	P-ZX	94.9	89.7	86.2	89.2	80.7	87.4
	FR	95.0	85.3	87.4	88.2	81.2	87.9
	DL	95.0	84.2	87.0	90.7	81.4	87.9
	DM,CPT	95.0	83.4	85.5	88.3	83.8	89.4

Cells with gray background indicate the results on the same adaptation and evaluation domains.

Table 9 Recall of all unknown words in each test set

Method	JNL	JPT	RCP	C-ZX	P-ZX	FR	DL	DM	CPT
BL	88.2	84.0	77.3	65.1	50.6	65.6	51.4	57.3	64.8
LF	89.1	83.9	78.7	75.6	64.4	53.5	55.2	57.2	59.3
LWP-S	90.3	89.1	82.2	69.0	53.1	65.3	51.9	58.4	67.2
LWP-T	91.9	89.4	84.1	80.5	73.6	58.6	60.6	59.5	67.5

curing in the training set. Table 9 shows the mean recall of three runs of each method on all unknown words in each test set. The proposed method, LWP-T, performed the best on 8 out of 9 domains and the improvements over the BL and the LF were +7.1 and +5.7 points on average. The performance of the LWP-S was in between that of the BL and the LWP-T on those 8 domains. The performance difference between the LWP-T and the LF suggests the importance of learning word information within the target domain’s context for accurate recognition of unknown words.

Next, we evaluated the performance on high frequency unknown word types in the JPT, C-ZX, and FR test sets. Table 10 shows the mean recall of three runs of the BL, LF, or LWP-T on each unknown word type. The proposed method recognized unknown words better than the baselines in most cases on JPT and C-ZX. This indicates that learning via the auxiliary labels in unlabeled data contributed to accurate recognition of these unknown words. In contrast, both LF and LWP-T had degraded performance on FR. This can be explained by the performance degradation on the words not contained in the lexicon used by both method, which corresponds to 5 of the top-10 unknown words in FR. This result suggests that the lexicon-based models are biased so that a character sequence without positive signals is not recognized as a word.

To analyze what words were not correctly recognized, we investigate the lexicon coverage rate (LCR), which indicates the percentage of word types contained in the lexicon,²³ for unknown word types. Table 11 shows the LCR for unknown word types recognized with high recall ($\geq 50\%$) and low recall ($< 50\%$) by the proposed method. The LCR was clearly lower for low recall words (−50 points on average) than high recall words. This matches the intuition that words not covered by lexicons are difficult to recognize, since there are no explicit clues to segment such character sequences as words. A direct solution is to construct and use lexicons with larger coverage by an automatic word acquisition method and/or manual annotation.

²³ For example, given a lexicon $\mathcal{V} = \{A, B, C\}$, the LCR for a set of word types $\{C, D, E, F\}$ is $1/4 = 25\%$.

Table 10 Recall of top-10 frequent unknown word types

(a) JPT

Word		In $\mathcal{L}_s \cup \mathcal{L}_t$	Freq	Recall		
				BL	LF	LWP_T
a	(a)	✓	205	75.8	95.2	100.0
前記	(above)	✓	139	89.2	100.0	100.0
電極	(electrode)	✓	111	99.1	99.1	100.0
膜	(membrane)	✓	91	65.2	74.7	89.7
モータ	(motor)	✓	76	62.7	99.4	100.0
センサ	(sensor)	✓	76	98.7	98.7	98.7
周波	(frequency)	✓	65	87.7	83.1	96.9
開口	(open)	✓	49	96.9	100.0	97.3
孔	(hole)	✓	49	97.3	81.7	95.9
コネクタ	(connector)	✓	47	81.0	100.0	100.0
Total			908	84.1	93.9	98.3

(b) C-ZX

Word		In $\mathcal{L}_s \cup \mathcal{L}_t$	Freq	Recall		
				BL	LF	LWP_T
张小凡	(person name)	✓	256	59.9	96.7	99.9
田不易	(person name)	✓	127	1.6	7.1	64.6
魔教	(demon)	✓	120	93.3	100.0	100.0
吸血	(haematophagy)	✓	105	99.7	87.0	100.0
苍松	(person name)	✓	94	34.4	100.0	100.0
田灵儿	(person name)	✓	91	83.5	100.0	100.0
老妖	(specter)	✓	83	75.5	86.0	100.0
鬼王	(person name)	✓	75	44.9	87.5	100.0
碧瑶	(person name)	✓	73	91.3	98.6	99.5
道人	(Taoist)	✓	70	9.0	95.7	100.0
Total			1,094	59.4	84.8	95.8

(c) FR

Word		In $\mathcal{L}_s \cup \mathcal{L}_t$	Freq	Recall		
				BL	LF	LWP_T
韩立	(person name)	✓	185	61.4	97.5	100.0
玄骨	(person name)		114	94.5	0.0	5.9
乌丑	(person name)		45	69.6	3.7	18.5
极阴	(person name)		40	42.5	0.0	0.0
蛮胡子	(person name)		39	28.2	0.0	0.0
血玉蜘蛛	(monster name)	✓	37	9.0	98.2	99.1
万天明	(person name)		36	12.0	0.0	0.9
虚天鼎	(weapon name)	✓	35	59.1	53.3	95.2
补天丹	(medicine name)	✓	32	16.7	97.9	100.0
冰焰	(skill name)	✓	29	100.0	57.5	44.8
Total			592	58.0	48.1	48.9

“In $\mathcal{L}_s \cup \mathcal{L}_t$ ” indicates whether the word is in the lexicon. “Freq” indicates the frequency of each word type in the test set.

Table 11 Lexicon coverage rate for unknown word types recognized with $\geq 50\%$ and $< 50\%$ recall by the proposed method LWP-T

	JNL	JPT	RCP	C-ZX	P-ZX	FR	DL	DM	CPT
Word types with recall $\geq 50\%$	83.9	72.3	86.9	88.6	96.9	89.5	92.3	83.5	61.0
Word types with recall $< 50\%$	60.8	45.1	68.7	41.7	26.0	22.2	27.7	9.0	13.2

5 Related Work

For both Chinese and Japanese languages, word segmentation has been traditionally addressed by applying statistical learning algorithms, such as maximum entropy (Uchimoto, Sekine, and Isahara 2001; Xue 2003), CRFs (Peng, Feng, and McCallum 2004; Kudo, Yamamoto, and Matsumoto 2004), structured perceptron (Zhang and Clark 2007, 2010), and logistic regression (Neubig et al. 2011).

Neural Network Models for Word Segmentation Various neural network architectures have been explored for Chinese word segmentation to reduce the burden of manual feature engineering. Specifically, character-based neural models have been developed to model the task as sequence labeling. Starting with work using feed-forward neural networks (Zheng et al. 2013; Mansur et al. 2013), more sophisticated architectures have also been used as main components of word segmentation models to derive effective features automatically; e.g., neural tensor networks (Pei et al. 2014), gated recursive neural networks (Chen et al. 2015a), LSTMs (Chen et al. 2015b; Ma, Ganchev, and Weiss 2018), CNNs (Chen, Qiu, and Huang 2017; Wang and Xu 2017), and SANs (Gan and Zhang 2019),

Word-based neural models have also been proposed. Typical models (Zhang et al. 2016; Cai and Zhao 2016; Cai et al. 2017; Yang, Zhang, and Liang 2019) sequentially determine whether or not to segment each character on the basis of word-level features and segmentation history, while keeping multiple segmentation candidates by beam search decoding.

There has been less work applying neural models on Japanese word segmentation than for Chinese. Morita, Kawahara, and Kurohashi (2015) integrated an RNN language model into a statistical Japanese morphological analysis framework, which simultaneously segments a sentence into words and predicts word features, such as POS and lemma. Kitagawa and Komachi (2018) applied a pure neural model based on an LSTM. Higashiyama, Utiyama, Sumita, Ideuchi, Oida, Sakamoto, and Okada (2019) proposed a BiLSTM-based model enhanced with word information using an attention mechanism. Tolmachev, Kawahara, and Kurohashi (2019) demonstrated that a BiLSTM or a SAN-based morphological analyzer relying only on character embeddings decreased

the model size by more than 95% compared to traditional dictionary-based models while achieving competitive performance.

Use of Linguistic Resources for Word Segmentation It has been demonstrated that the use of linguistic resources, such as unlabeled/partially-labeled data and lexicons, has achieved robust performance for out-of-domain texts. Well-known techniques using unlabeled data include self-training (Liu and Zhang 2012) and statistical features (Wu et al. 2014; Sudoh, Nagata, Mori, and Kawahara 2014) such as accessor variety and branching entropy. Punctuation and hyperlink information can be regarded as natural annotation. Text with such information was viewed as partially-labeled data (Jiang, Sun, Lü, Yang, and Liu 2013; Zhang, Li, He, Wang, and Sun 2013; Liu et al. 2014). Distantly-supervised data generated from unlabeled data and lexicons was also used as partially-labeled data (Liu et al. 2014; Zhao et al. 2018). Another well-known technique based on lexicons is a lexicon feature that indicates occurrence of lexicon entries (Neubig et al. 2011; Zhang et al. 2014; Wu et al. 2014).

As common practice in recent neural models, large unlabeled text has been used to pre-train character/word embeddings (Zheng et al. 2013; Chen et al. 2015b; Zhang et al. 2016). The work mainly focused on improvements of in-domain performance by using general texts such as news or Wikipedia. On the other hand, Zhou, Yu, Zhang, Huang, Dai, and Chen (2017) and Ye et al. (2019) proposed word segmentation-oriented training methods of character or word embeddings. Both showed that their (pre-) trained embeddings learned from target domain text contributed to performance improvements on the target domains. Wang, Cai, Li, Xu, Zhao, and Si (2019) integrated auto-segmented label information by an unsupervised segmenter into a neural segmentation model to boost performance of in-domain word segmentation.

Some recent work explored neural models incorporating unlabeled data and/or lexicons in different manners and showed their improved performance for target domains such as scientific literature, novels, and social media. Zhang et al. (2018) proposed a character-based BiLSTM model integrated with discrete lexicon features and added a target lexicon when decoding text in target domains. Zhao et al. (2018) proposed a character-based BiLSTM model that made use of unlabeled and partially-labeled data in target domains. They combined a segmentation model with a character-level language model learned from unlabeled data and trained the model with a modified loss function to handle partially-labeled data. Liu, Wu, Wu, Huang, and Xie (2019) proposed a character-based CNN model integrated with a regularization loss based on a lexicon so that the model’s predictions for unlabeled sentences included more words in the lexicon. Gan and Zhang (2019) proposed a character-based SAN model enhanced with word embeddings. They used a target word-POS lexicon for a domain adaptation technique that used POS embeddings,

instead of word embeddings, for domain-specific words whose embeddings were non-available.

6 Conclusion

In this work, we proposed a novel method using unlabeled data and lexicons for cross-domain word segmentation. To recognize unknown words not occurring in source domain training data, we incorporated lexical knowledge into a neural character-based segmenter as an auxiliary prediction task to identify word occurrences in unlabeled sentences. We conducted domain adaptation experiments on Japanese and Chinese datasets with various target domain test sets, including science and technology, recipes, and novels. The experimental results demonstrated that our auxiliary task improved performance for target domains by 3.2 F1 points on average over the baseline BiLSTM segmenter, while achieving similar or better performance for source and other domains. Additionally, compared with existing Japanese and Chinese word segmenters, our method achieved better or competitive performance.

Acknowledgement

We would like to thank our anonymous reviewers for their constructive comments. We used the Cookpad dataset provided by Cookpad Inc. via IDR Dataset Service of the National Institute of Informatics.

References

- Cai, D. and Zhao, H. (2016). “Neural Word Segmentation Learning for Chinese.” In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 409–420.
- Cai, D., Zhao, H., Zhang, Z., Xin, Y., Wu, Y., and Huang, F. (2017). “Fast and Accurate Neural Word Segmentation for Chinese.” In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 608–615.
- Chen, X., Qiu, X., and Huang, X. (2017). “A Feature-enriched Neural Model for Joint Chinese Word Segmentation and Part-of-speech Tagging.” In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3960–3966.
- Chen, X., Qiu, X., Zhu, C., and Huang, X. (2015a). “Gated Recursive Neural Network for Chinese Word Segmentation.” In *Proceedings of the 53rd Annual Meeting of the Association for*

- Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, Vol. 1, pp. 1744–1753.
- Chen, X., Qiu, X., Zhu, C., Liu, P., and Huang, X. (2015b). “Long Short-term Memory Neural Networks for Chinese Word Segmentation.” In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1197–1206.
- Collobert, R. and Weston, J. (2008). “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning.” In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pp. 160–167.
- Den, Y., Ogiso, T., Ogura, H., Yamada, A., Minematsu, N., Uchimoto, K., and Koiso, H. (2007). “The Development of an Electronic Dictionary for Morphological Analysis and Its Application to Japanese Corpus Linguistics.” *Japanese Linguistics*, **22**, pp. 101–123.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186.
- Emerson, T. (2005). “The 2nd International Chinese Word Segmentation Bakeoff.” In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing*.
- Gan, L. and Zhang, Y. (2019). “Investigating Self-attention Network for Chinese Word Segmentation.” *Computing Research Repository*, **arXiv:1907.11512**.
- Gillick, L. and Cox, S. J. (1989). “Some Statistical Issues in the Comparison of Speech Recognition Algorithms.” In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 532–535.
- Higashiyama, S., Utiyama, M., Sumita, E., Ideuchi, M., Oida, Y., Sakamoto, Y., and Okada, I. (2019). “Incorporating Word Attention into Character-Based Word Segmentation.” In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 2699–2709.
- Hochreiter, S. and Schmidhuber, J. (1997). “Long Short-term Memory.” *Neural Computation*, **9** (8), pp. 1735–1780.
- Huang, Z., Xu, W., and Yu, K. (2015). “Bidirectional LSTM-CRF Models for Sequence Tagging.” *Computing Research Repository*, **arXiv:1508.01991**.
- Jiang, W., Sun, M., Lü, Y., Yang, Y., and Liu, Q. (2013). “Discriminative Learning with Natural Annotations: Word Segmentation as a Case Study.” In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 761–769.
- Kitagawa, Y. and Komachi, M. (2018). “Long Short-term Memory for Japanese Word Segmen-

- tation.” In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation (PACLIC)*, pp. 279–288.
- Kudo, T., Yamamoto, K., and Matsumoto, Y. (2004). “Applying Conditional Random Fields to Japanese Morphological Analysis.” In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 230–237.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). “Gradient-based Learning Applied to Document Recognition.” *Proceedings of the IEEE*, **86** (11), pp. 2278–2324.
- Liu, J., Wu, F., Wu, C., Huang, Y., and Xie, X. (2019). “Neural Chinese Word Segmentation with Lexicon and Unlabeled Data via Posterior Regularization.” In *The World Wide Web Conference (WWW)*, pp. 3013–3019.
- Liu, Y. and Zhang, Y. (2012). “Unsupervised Domain Adaptation for Joint Segmentation and POS-tagging.” In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pp. 745–754.
- Liu, Y., Zhang, Y., Che, W., Liu, T., and Wu, F. (2014). “Domain Adaptation for CRF-based Chinese Word Segmentation using Free Annotations.” In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 864–874.
- Ma, J., Ganchev, K., and Weiss, D. (2018). “State-of-the-art Chinese Word Segmentation with Bi-LSTMs.” In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4902–4908.
- Maekawa, K., Yamazaki, M., Ogiso, T., Maruyama, T., Ogura, H., Kashino, W., Koiso, H., Yamaguchi, M., Tanaka, M., and Den, Y. (2014). “Balanced Corpus of Contemporary Written Japanese.” *Language Resources and Evaluation*, **48** (2), pp. 345–371.
- Mansur, M., Pei, W., and Chang, B. (2013). “Feature-based Neural Language Model and Chinese Word Segmentation.” In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCINLP)*, pp. 1271–1277.
- Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). “Distant Supervision for Relation Extraction without Labeled Data.” In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCINLP)*, pp. 1003–1011.
- Mori, S., Ogura, H., and Sasada, T. (2014). “A Japanese Word Dependency Corpus.” In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pp. 753–758.
- Morita, H., Kawahara, D., and Kurohashi, S. (2015). “Morphological Analysis for Unsegmented Languages using Recurrent Neural Network Language Model.” In *Proceedings of the 2015*

- Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2292–2297.
- Neubig, G., Nakata, Y., and Mori, S. (2011). “Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis.” In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pp. 529–533.
- Pei, W., Ge, T., and Chang, B. (2014). “Max-Margin Tensor Neural Network for Chinese Word Segmentation.” In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Vol. 1, pp. 293–303.
- Peng, F., Feng, F., and McCallum, A. (2004). “Chinese Segmentation and New Word Detection using Conditional Random Fields.” In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pp. 562–568.
- Qiu, L., Shi, L., and Wang, H. (2015). “Construction of Multi-Domain Chinese Dependency Treebanks and a Study on Factors Influencing the Statistical Parsing.” *Journal of Chinese Information Processing*, **29** (5), pp. 69–75.
- Qiu, L. and Zhang, Y. (2015). “Word Segmentation for Chinese Novels.” In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2440–2446.
- Sudoh, K., Nagata, M., Mori, S., and Kawahara, T. (2014). “Japanese-to-English Patent Translation System based on Domain-adapted Word Segmentation and Post-ordering.” In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas (AMTA)*, pp. 234–248.
- Tolmachev, A., Kawahara, D., and Kurohashi, S. (2019). “Shrinking Japanese Morphological Analyzers with Neural Networks and Semi-supervised Learning.” In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 2744–2755.
- Uchimoto, K., Sekine, S., and Isahara, H. (2001). “The Unknown Word Problem: a Morphological Analysis of Japanese Using Maximum Entropy Aided by a Dictionary.” In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). “Attention is All You Need.” In *Advances in Neural Information Processing Systems (NIPS)*, pp. 5998–6008.
- Wang, C. and Xu, B. (2017). “Convolutional Neural Network with Word Embeddings for Chinese Word Segmentation.” In *Proceedings of The 8th International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 163–172.
- Wang, X., Cai, D., Li, L., Xu, G., Zhao, H., and Si, L. (2019). “Unsupervised Learning Helps

- Supervised Neural Word Segmentation.” In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 33, pp. 7200–7207.
- Wu, G., He, D., Zhong, K., Zhou, X., and Yuan, C. (2014). “Leveraging Rich Linguistic Features for Cross-domain Chinese Segmentation.” In *Proceedings of The 3rd CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pp. 101–107.
- Xue, N. (2003). “Chinese Word Segmentation as Character Tagging.” *International Journal of Computational Linguistics & Chinese Language Processing (IJCLCLP)*, Volume 8, Number 1, February 2003: Special Issue on Word Formation and Chinese Language Processing, 8 (1), pp. 29–48.
- Yang, J., Zhang, Y., and Dong, F. (2017). “Neural Word Segmentation with Rich Pretraining.” In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 839–849.
- Yang, J., Zhang, Y., and Liang, S. (2019). “Subword Encoding in Lattice LSTM for Chinese Word Segmentation.” In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 2720–2725.
- Ye, Y., Li, W., Zhang, Y., Qiu, L., and Sun, J. (2019). “Improving Cross-domain Chinese Word Segmentation with Word Embeddings.” In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 2726–2735.
- Zaremba, W., Sutskever, I., and Vinyals, O. (2015). “Recurrent Neural Network Regularization.” In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.
- Zhang, L., Li, L., He, Z., Wang, H., and Sun, N. (2013). “Improving Chinese Word Segmentation on Micro-blog using Rich Punctuations.” In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 177–182.
- Zhang, M., Zhang, Y., Che, W., and Liu, T. (2014). “Type-supervised Domain Adaptation for Joint Segmentation and POS-tagging.” In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 588–597.
- Zhang, M., Zhang, Y., and Fu, G. (2016). “Transition-based Neural Word Segmentation.” In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vol. 1, pp. 421–431.
- Zhang, Q., Liu, X., and Fu, J. (2018). “Neural Networks Incorporating Dictionaries for Chinese Word Segmentation.” In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 5682–5689.

- Zhang, Y. and Clark, S. (2007). “Chinese Segmentation with a Word-Based Perceptron Algorithm.” In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pp. 840–847.
- Zhang, Y. and Clark, S. (2010). “A Fast Decoder for Joint Word Segmentation and POS-Tagging Using a Single Discriminative Model.” In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 843–852.
- Zhang, Y. and Clark, S. (2011). “Syntactic Processing Using the Generalized Perceptron and Beam Search.” *Computational Linguistics*, **37** (1), pp. 105–151.
- Zhao, L., Zhang, Q., Wang, P., and Liu, X. (2018). “Neural Networks Incorporating Unlabeled and Partially-labeled Data for Cross-domain Chinese Word Segmentation.” In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4602–4608.
- Zheng, X., Chen, H., and Xu, T. (2013). “Deep Learning for Chinese Word Segmentation and POS Tagging.” In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 647–657.
- Zhou, H., Yu, Z., Zhang, Y., Huang, S., Dai, X., and Chen, J. (2017). “Word-Context Character Embeddings for Chinese Word Segmentation.” In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 760–766.

Shohei Higashiyama: received a B.S degree in engineering and an M.S. degree in system informatics from Kobe University in 2012 and 2014, respectively. He was a researcher at NEC Corporation from 2014 to 2019. He is currently a Technical Researcher at the National Institute of Information and Communications Technology (NICT) and a Ph.D. student at Nara Institute of Science and Technology (NAIST). His main research interests are natural language processing and machine learning.

Masao Utiyama: received a B.S. in Computer Science from the University of Tsukuba, Japan, in 1992, an M.S. in Computer Science from the University of Tsukuba, Japan, in 1994, a Ph.D. in Engineering from the University of Tsukuba, Japan, in 1997. From 1997 to 1999, he was a Research Associate at Shinshu University, Japan. He has been a member of the National Institute of Information and Communications Technology, Japan, since 1999, and is currently an Executive Researcher at NICT.

Yuji Matsumoto: is currently a team leader at RIKEN Center for Advanced

Intelligence Project (AIP). He received his M.S. and Ph.D. degrees in information science from Kyoto University in 1979 and in 1990, respectively. He joined Machine Inference Section of Electrotechnical Laboratory in 1979. He has been an academic visitor at Imperial College of Science and Technology, the deputy chief of First Laboratory at ICOT, an associate professor at Kyoto University, and a professor at Nara Institute of Science and Technology. His main research interests are natural language understanding and machine learning. He is an ACL fellow and a fellow of Information Processing Society of Japan.

Taro Watanabe: received the B.E. and M.E. degrees in information science from Kyoto Univ., Kyoto, Japan in 1994 and 1997, respectively, and obtained the Master of Science degree in language and information technologies from the School of Computer Science, Carnegie Mellon University in 2000. In 2004, he received the Ph.D. in informatics from Kyoto Univ., Kyoto, Japan. After working as a researcher at ATR, NTT and NICT, and as a software engineer at Google, he is a professor at Nara Institute of Science and Technology starting from 2020. His research interests include natural language processing, machine learning and machine translation.

Eiichiro Sumita: received his Ph.D. in Engineering from Kyoto University in 1999, and a master's and bachelor's in Computer Science from the University of Electro-Communications in 1982 and 1980, respectively. He is now in the National Institute of Information and Communication Technology (NICT), its fellow and the associate director-general of Advanced Speech Translation Research and Development Promotion Center (ASTREC). His research interests cover machine translation and e-learning. He is a co-recipient of the Maejima Hisoka Prize in 2013, the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology, Prizes for Science and Technology in 2010, and the AAMT Nagao Award in 2007.

(Received May 1, 2020)

(Accepted June 17, 2020)