Doctoral Dissertation

Word Segmentation and Lexical Normalization for Unsegmented Languages

Shohei Higashiyama

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

Superviser: Prof. Taro Watanabe
Natural Language Processing Laboratory
(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

Shohei Higashiyama

Thesis Committee:
Supervisor Taro Watanabe
(Professor, Division of Information Science)
Satoshi Nakamura
(Professor, Division of Information Science)
Hiroyuki Shindo
(Affiliate Associate Professor, Division of Information Science)
Yuji Matsumoto
(Team Leader, RIKEN AIP)

Word Segmentation and Lexical Normalization for Unsegmented Languages[*]

Shohei Higashiyama

Abstract

Word segmentation is a fundamental technology in natural language processing in unsegmented languages such as Japanese and Chinese. Although accurate segmentation is required to prevent error propagation to downstream tasks, proper approaches can differ according to the characteristics and available linguistic resources in text domains. For general (and any other) domains, a disambiguation method for ambiguous word boundaries is necessary. For specialized domains, the use of unlabeled data and lexicons is effective in recognizing domain-specific words. For user-generated text domains, lexical normalization of nonstandard words is important because these words can degrade the performance of word segmentation and downstream tasks. In this thesis, we present our work on four problems to achieve accurate Japanese and Chinese word segmentation in various domains and Japanese lexical normalization in user-generated text.

First, aiming to improve the performance of in-domain semi-supervised word segmentation through a better disambiguation process of word boundaries, we propose a neural character-based model that learns the importance of multiple candidate words for each character.

Second, for cross-domain word segmentation where few labeled data are available in target domains, we propose a model that learns the occurrences of lexical words in unlabeled target sentences, together with segmentation label information from labeled source sentences.

In addition, we tackle two problems in Japanese word segmentation and lexical normalization of user-generated text, i.e., the lack of public evaluation data and

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

the lack of training data. For the former problem, we have constructed a public Japanese evaluation dataset annotated with morphological, normalization, and word category information to enable a detailed evaluation of segmentation and normalization systems. For the latter problem, we propose methods for generating pseudo-labeled data using segmented sentences and standard and nonstandard word variant pairs, and a neural text editing model for joint Japanese word segmentation, part-of-speech tagging, and lexical normalization to efficiently train on generated pseudo-labeled data.

Through our work, we demonstrated that our proposed segmentation methods for different domain types achieved accurate performance in various domains. In addition, our Japanese lexical normalization method can be a good baseline for developing more practical systems in the future, and our evaluation dataset can be a useful benchmark for comparing and analyzing existing and future systems.

Keywords:

word segmentation, lexical normalization, part-of-speech tagging, Japanese morphological analysis, sequence labeling, user-generated text

Acknowledgements

初めに，奈良先端科学技術大学院大学（奈良先端大）教授 渡辺太郎先生に感謝申し上げます．2020年4月のご着任から2年間，指導教員としてご指導いただきました．日頃のミーティングでのご助言や論文執筆時の丁寧なレビューをしていただいたおかげで，国際会議・論文誌への投稿論文が順調に採択に至り，博士課程の研究を円滑に進めることができました．また，本学位論文の執筆にあたり数多くのご指摘をいただきました．

理化学研究所の松本裕治先生に感謝申し上げます．私の入学年度は，松本先生が奈良先端大教授をご在任された最終年度であったにもかかわらず，旧松本研究室への配属を快く受け入れてくださいました．1年間のご指導の間に様々なご助言をいただき，単語分割に加えて語彙正規化を行うという博士課程での研究の方向性を具体化することができました．また，学外からの審査委員として本学位論文を審査いただきました．

副指導教員ならびに本学位論文の審査委員をご担当いただいた奈良先端大教授 中村哲先生と特任准教授 進藤裕之先生に感謝申し上げます．予備審査でいただいたご指摘・ご質問を基に，重要な現象に対する提案手法の性能等について，本論文における分析・議論をより深めることができました．

情報通信研究機構 先進的翻訳技術研究室の隅田英一郎室長に感謝申し上げます．2017年5月に私が同機構に着任して以降，研究生活全般にわたるご支援をいただき，本学在学中も安心して研究に邁進することができました．同研究室に受け入れていただいて以降，私の研究人生は一層充実したものとなりました．

同機構の内山将夫さんに感謝申し上げます．私の着任時から継続して研究を見ていただき，数々のご助言をいただいたことで，NAACL-HLT 2019での論文採択を始めとする研究成果を挙げることができました．また，博士課程進学を何度か勧めてくださったことで，学位取得に向けた意欲が徐々に高まり，進学を決意するに至りました．

同機構の同僚・元同僚の皆様に感謝申し上げます．特に，土屋雅稔さん（豊橋技術科学大学），出内将夫さん，田中英輝さん，藤田篤さん，今村賢治さんには，日頃のミーティングなどで多数の有益なコメントを頂き，本学在学中の研究をより発展させることができました．また，藤田篤さん，Wang Rui さん（現在Shanghai Jiao Tong University）には，同機構での私の一つ目の研究業績となった単語分割の論文をレビューしていただき，難関国際会議採択に至る論文の質へ高めることができました．

私が安心して日々の研究に取り組めたのは，同機構同研究室のアシスタントの皆様，ならびに奈良先端大渡辺研究室の秘書 北川祐子さんの研究事務に関するご支援のおかげです．感謝申し上げます．

情報通信研究機構在職および奈良先端大在学に至るまでには，多くの方々からご支援いただきました．NEC中央研究所旧テキストマイニングチームおよび旧自動推論テクノロジーグループの皆様． 特に，同社在職中，直属上司として，企業研究者としての基礎能力を鍛えていただいた定政邦彦さん． 神戸大学在学時にご指導いただいた上原邦昭先生（現在大阪学院大学），関和広先生（現在甲南大学），Mathieu Blondelさん（現在Google Research）．神戸大学にて共に切磋琢磨した同期の皆さん．以上の方々に感謝申し上げます． そして，これまでに私の研究に関して建設的なコメントをくださったすべての方に感謝申し上げます．

奈良先端大より，社会人学生への奨学支援として，在学期間中の授業料全額免除ならびに奨学一時金の給付をしていただきました．ここに感謝申し上げます．

Contents

List of Figures

List of Tables

1. Introduction

Word segmentation is the task of segmenting an unsegmented sentence into words, and a fundamental technology for downstream natural language processing (NLP) tasks in languages written without explicit word delimiters, such as Japanese and Chinese. This task is a prerequisite step for linguistic analysis, such as dependency parsing [1] and predicate argument structure analysis [97], which require human-understandable segments, i.e., words. In addition, word information is useful as a feature, an auxiliary task, or an intermediate unit to be split into subwords for application-oriented tasks, such as named entity recognition (NER) [34, 80] and machine translation (MT) [115, 149]. Both types of tasks require accurate word segmentation to prevent error propagation.

Major problems in word segmentation include segmentation ambiguity, domain-specific words,[1] and nonstandard words. The necessity to deal with these phenomena differs according to the characteristics and available linguistic resources in text domains. First, segmentation ambiguity is an inevitable problem in any domain, including general domains where manually labeled data are available. For example, a Japanese phrase 日本人 can be segmented into 日本|人 *nihon jin* ‘Japanese person’ or 日|本人 *hi hon-nin* ‘day, the person.’ A segmentation system needs to predict the former as more probable segmentation, although the latter can also be appropriate depending on the surrounding context.[2] Second, domain-specific words are problematic in non-general domains because labeled data containing such words are usually unavailable. For example, パープレキシティ ‘perplexity’ (a term used in NLP) and 捏和/ねっ和 *nekka* ‘kneading’ (a

[1] Domain-specific words are often unknown words that do not occur in a training corpus or a given lexicon for a segmentation system.

[2] A sentence fragment 明日本人 can be segmented into 明日|本人 *ashita honnin* ‘tomorrow, the person,’ whereas 未明日本人 can be segmented into 未明|日本|人 *mimē nihon jin* ‘wee hours, Japanese person.’

term used in the chemical and manufacturing industries) rarely occur in general text. Third, nonstandard words frequently occur in user-generated text (UGT), such as social media and blog posts. For example, すごい/凄い *sugoi* 'awesome' is occasionally written as nonstandard forms such as すごーい *sugōi* and スッゲエェ *suggē*. In addition to identifying nonstandard words, lexical normalization of nonstandard words is important in UGT domains because these words can degrade the performance of word segmentation and downstream tasks.

In this thesis, to achieve accurate Japanese and Chinese word segmentation in various domains and Japanese lexical normalization in UGT, we propose proper segmentation and normalization approaches according to three domain types: general, specialized, and UGT domains.

For word segmentation in general domains such as news articles, manually annotated corpora have been constructed and published [28, 58, 72, 152], and many (semi-) supervised neural network models have been actively developed during this decade [8, 14, 53, 172]. However, although candidate word information for characters in a sentence is beneficial for disambiguating word boundaries, limited effort has been devoted to leveraging word information in character-based models, which is a dominant approach. In Chapter 3, aiming to achieve better performance in in-domain semi-supervised word segmentation, we propose a neural character-based model that learns and distinguishes the importance of candidate words in different context via an attention mechanism.

In specialized domains such as technical fields or specific text topics, performance degradation is a severe problem because of a small amount of labeled data and the existence of domain-specific words. A promising approach to this problem is to combine source domain labeled data and target domain resources such as lexicons and unlabeled data, which can be collected or constructed more easily than manually labeled data. Although existing methods include lexicon features [92, 165] and distant supervision [68, 171], the former methods may not sufficiently adapt to target domains, and noisy pseudo-labels from the latter methods may hurt model performance. In Chapter 4, we propose a method that integrates knowledge from these resources into a neural segmentation model for cross-domain word segmentation. With the help of our auxiliary prediction task, a model learns the occurrences of lexical entries in unlabeled target sentences, together with segmentation label information from source labeled sentences.

For UGT, public annotated corpora [35, 159] have promoted research on English lexical normalization [7, 49, 78]. However, there are two problems with Japanese lexical normalization, i.e., the lack of public evaluation data and the lack of training data. Previous work on Japanese lexical normalization [51, 108, 112] have developed and evaluated systems using individual in-house data; thus, it is difficult to compare the performance and issues of different systems. For the former problem, in Chapter 5, we present a public Japanese evaluation dataset that we constructed and annotated with morphological and normalization information, along with category information of UGT-specific words for a detailed evaluation.

For the latter problem, in Chapter 6, we propose methods for generating pseudo-labeled data using (auto-) segmented sentences and standard and non-standard word variant pairs constructed based on lexical knowledge, i.e., a dictionary and hand-crafted rules. In addition, to efficiently train on generated pseudo-labeled data, we propose a neural text editing model designed for joint Japanese word segmentation, part-of-speech (POS) tagging, and lexical normalization.

This thesis makes the following contributions. (1) Our proposed segmentation methods for the three domain types can be effective options for achieving accurate word segmentation and downstream tasks in various domains. (2) Our Japanese lexical normalization method can be a good baseline for developing more practical systems in the future. (3) Our evaluation dataset can be a useful benchmark for comparing and analyzing existing and future systems.

The rest of this thesis is organized as follows. Chapter 2 describes the background on word segmentation and lexical normalization as well as our baseline model architecture for those tasks. Chapter 3–6 describe our work on the above problems. Chapter 7 summarizes this thesis and discusses future directions.[3]

[3] Major parts of our four studies were published in the following conference proceedings and journals: Proceedings of NAACL-HLT 2019 [39], Journal of Natural Language Processing, Vol. 27, No. 3 [38, 40], Proceedings of NAACL-HLT 2021 [42], and Proceedings of W-NUT 2021 [41].

2. Preliminaries and Background

2.1. Word Segmentation

2.1.1. Task Definition

Word segmentation is the task of segmenting an unsegmented sentence into words. We treat word segmentation as a character-level sequence labeling task. Labeled data $\mathcal{D} = \{(\boldsymbol{x}, \boldsymbol{t})\}$ for this task is a set of pairs of a sentence $\boldsymbol{x} = \boldsymbol{x}_{1:n} = (x_1, \cdots, x_n)$ and its gold label sequence $\boldsymbol{t} = \boldsymbol{t}_{1:n} = (t_1, \cdots, t_n)$, 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 $\boldsymbol{x}$, a segmentation model is required to predict a label sequence $\boldsymbol{y} = \boldsymbol{y}_{1:n} = (y_1, \cdots, y_n) \in \mathcal{T}^n$. We employ a 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 [153].[4]

Figure 2.1 shows an example sentence $\boldsymbol{x}$ and its gold label sequence $\boldsymbol{t}$. The label sequence $\boldsymbol{t}$ indicates that $\boldsymbol{x}$ is segmented into five words: テキスト *tekisuto* 'text,' の *no* GEN, 分割 *bunkatsu* 'segmentation,' と *to* 'and,' 正規 *sēki* 'regular/normal,' and 化 *ka* '-ization.'

i	1	2	3	4	5	6	7	8	9	10	11
x_i	テ	キ	ス	ト	の	分	割	と	正	規	化
t_i	`B`	`I`	`I`	`E`	`S`	`B`	`E`	`S`	`B`	`E`	`S`

Table 2.1. Segmentation label sequence $\boldsymbol{t}$ of a sentence $\boldsymbol{x}$ ="テキストの分割と正規化" *tekisuto no bunkatsu to sēkika* 'Segmentation and normalization of text.'

[4] For post-processing of predicted label sequences that violate the tagging scheme, such as "$\cdots$`IB`$\cdots$" and "$\cdots$`EI`$\cdots$", we adopt the same correction rules as the script used in the CoNLL-2003 shared task (`https://www.clips.uantwerpen.be/conll2003/ner/bin/conlleval`).

2.1.2. Segmentation Criteria

A word is a syntactic and semantic unit composing a sentence or utterance. Word segmentation aims to obtain words as linguistic units and has been treated as the first step in NLP. However, it is difficult to identify word boundaries or define what a word is in unsegmented languages such as Japanese and Chinese.

For Japanese, word segmentation, POS tagging, and lemmatization have been researched together as a joint task called Japanese morphological analysis (MA) [57, 131]. Some segmentation criteria, along with POS tag sets, have been designed and used in NLP and corpus linguistics research. Kyoto University developed the Kyoto University text corpus [58] and Japanese MA systems, JUMAN and JUMAN++. The JUMAN POS tag set for the corpus and systems was defined based on Masuoka and Takubo's Japanese grammar book [76]. The National Institute for Japanese Language and Linguistics (NINJAL) defined the short unit word (SUW) criterion for consistent segmentation and the long unit word (LUW) criterion for various compound words. These criteria have been used in the corpora of NINJAL, including the corpus of spontaneous Japanese [71] and the balanced corpus of contemporary written Japanese (BCCWJ) [72], and an MA dictionary called UniDic [24]. A recent Japanese MA system, Sudachi [129], adopted multi-granular segmentation allowing a user to select a preferable unit from three units: short, middle, and named entity (NE) units.

In addition, for Chinese, several word segmentation corpora with their own criteria have been developed. The Penn Chinese Treebank Project developed the Penn Chinese Treebank [147, 148], which is a Mandarin Chinese text corpus with word boundaries, POS tags, and syntactic bracketing. The first international Chinese word segmentation bakeoff [120][5] was held in 2003, and additional bakeoffs have since been held. For the participants of the bakeoffs, annotated corpora based on different criteria were provided as official training and test datasets from several research institutes. Both the Penn Chinese Treebank and the datasets of the bakeoffs have been used as standard benchmarks for Chinese word segmentation in much subsequent research [14, 70, 100, 172].

Previous work [120, 129, 145] has discussed that segmentation criteria with different granularity can be appropriate in different downstream tasks. For example,

[5] `http://sighan.cs.uchicago.edu/bakeoff2003/`

a short unit is beneficial for improving the recall of search systems, whereas a long unit is useful for defining the syntactic dependency between words. In this thesis, we do not assume a specific criterion and evaluate the proposed word segmentation methods on multiple Japanese and Chinese datasets with different criteria.

2.1.3. Subword Segmentation.

In word-level processing in traditional NLP pipelines, a large number of word types and the long-tail distribution cause infrequent and unknown words, which often degrade the downstream task performance. Alternative character-level processing increases the sentence length and thus requires the capability to learn long-term dependency and long processing time. Recent end-to-end neural text processing, particularly text generation such as MT [84] and dialogue generation [124], has actively adopted subword segmentation methods [56, 114, 115] to achieve a reasonable balance between the vocabulary size and sentence length. Because of unsupervised data-driven segmentation, subword tokens are often incomprehensible for humans; subword boundaries do not necessarily match those of words or morphemes. This may not be a problem when the main interest of users is achieving high accuracy in downstream tasks.

By contrast, word segmentation is a prerequisite step, even in neural text processing, to obtain a linguistic unit for syntactic and semantic analysis tasks, such as dependency parsing [1], predicate argument structure analysis [97], and coreference resolution [118]. In addition, word boundary information can be useful for information extraction and knowledge base construction, such as NER [34, 80] and entity relation extraction [15], as well as NLP applications for supporting human activities, such as grammatical error correction for language learners [54], automated proofreading and term correction [130, 155], and text mining in specialized domains [10, 109].

2.1.4. Linguistic Phenomena Related to Word Segmentation

Derivative and Compound Word. In terms of word formation, words are organized into a simple word (単純語), a derivative word (派生語), and a compound word (複合語) [95]. A simple word is formed from a single morpheme. A derivative word is formed by adding affixes to a word. For example, a Japanese noun 強さ *tsuyosa* '(degree of) strength' is formed from the base 強 *tsuyo* of an adjective 強い *tsuyoi* 'strong' and a nominal suffix さ *sa*. Similarly, a Japanese noun 強み *tsuyomi* 'strong point' is formed from 強 and a nominal suffix み *mi*. A compound word is formed from more than two words. For example, a Japanese word 新米 *shimmai* 'new rice' is formed from 新 *shin* 'new' and 米 *kome/mai* 'rice', and a made-in-Japan English word ユニットバス 'modular bath' is formed from ユニット 'unit' and バス 'bath.' However, it is non-trivial to treat a derivative or compound word as a single word or as two or more words. For example, 強さ is segmented as 強|さ, and 強み is treated as a single word based on SUW, whereas both words are segmented into two separate words based on the JUMAN POS tag set. By contrast, ユニットバス is segmented into ユニット|バス based on SUW, but is treated as a single word based on the JUMAN POS tag set.

Although it may be preferable to treat long compound words and NEs as single tokens according to downstream tasks, we do not address this problem in our work because it can be achieved through an additional chunking process [55, 111, 136]. We expect that our (semi-) supervised models learn whether to segment each derivative and compound word into multiple tokens based on given training corpora.

Polysemic Word. In terms of semantics, a word is monosemic (単義) when it has a single meaning, and a word is polysemic (多義) when it has multiple meanings. The most basic words that are frequently used are considered polysemic [86]. For example, a Japanese word 手 *te* has several meanings, including 'hand,' 'handle of an equipment,' and 'means or way,' and 新米 *shimmai* has (at least) two meanings: 'new rice' and 'novice.' A polysemic word can ambiguate the meaning of a phrase or sentence: 新米販売員 *shimmai hambai-in* can be interpreted as 'new rice salesperson' or 'novice salesperson.' The polysemy of

	Phrase	Segmentation	POS of “米”
(1a)	日本米 *nihon mai* ‘Japanese rice’	日本|米	Suffix
(1b)	無洗米 *musen mai* ‘no-wash rice’	無洗|米	Suffix
(2)	米政府 *bei sēfu* ‘the United States Government’	米|政府	Noun
(3)	二百米 *nihyaku mētoru* ‘two hundred meters’	二百|米	Counter word

Table 2.2. Examples of homographs. Segmentation is based on SUW.

words, however, does not usually affect word segmentation results; namely, segmentation of the same word (e.g., 新米) is the same regardless of the meaning of the word (e.g., ‘new rice’ or ‘novice’). On the basis of this consideration, we do not explicitly model the polysemy in our work.

Homograph. A homograph (同形異義語) is a word written with the same form as other words but differing in meaning and occasionally pronunciation. For example, a Japanese words 米 *mai/bē/mētoru* in phrases (1a) and (1b), (2), and (3) in Table 2.2[6] mean ‘rice,’ ‘America,’ and ‘meter,’ respectively, whereas a word segmentation system only needs to recognize them as single-character words regardless of their meanings. However, homographs occasionally cause segmentation ambiguity, as discussed below. Notably, it is necessary to discriminate these homographs in POS tagging or lemmatization.

Segmentation Ambiguity. Segmentation ambiguity refers to a phenomenon that a character sequence can be segmented into multiple different word sequences. For example, a Japanese phrase 米人口 can be segmented into 米|人口 *bē jinkō* ‘the population of America’ or 米人|口 *bējin kuchi/kō* ‘American person, mouth.’ Although the former is much more likely in many cases, this ambiguity makes word segmentation difficult. Sun and T’sou [126] described similar ambiguous word boundaries in Chinese. A phrase 米原発 is a more difficult example, and can be segmented into 米|原発 *bē gempatsu* ‘nuclear power (plant) in America’ or 米原|発 *maibara hatsu* ‘from Maibara’, although both are meaningful and proper segmentation is context-dependent. 米|原発|産業 *bē gempatsu*

[6] The formal POS tag names of a suffix, noun, and counter word of SUW are “接尾辞-名詞的-一般,” “名詞-普通名詞-一般,” and “名詞-普通名詞-助数詞可能,” respectively.

sangyō ‘the nuclear power generation industry in America’ and 米原|発|大垣|行 *maibara hatsu ōgaki yuki* ‘from Maibara to Ōgaki’ are likely unique and proper segmentation (based on SUW).

Including our baseline model in §2.3, word segmentation systems solve the segmentation ambiguity to a certain extent based on the surrounding context of the characters of interest. In addition, we expect our segmentation methods to deal more effectively with the ambiguity problem. The proposed method in Chapter 3 explicitly models the interaction between characters and candidate words via an attention mechanism. The proposed method in Chapter 4 learns the position information of characters in possible lexical words via an auxiliary prediction task.

Unknown Word. An unknown or out-of-vocabulary (OOV) word indicates a word that is not included in a specified vocabulary, which usually corresponds to all words in a training corpus and/or a lexicon [135]. Typical examples of unknown words are technical terms, proper nouns, and new words. Unknown word processing is a long-standing problem in Japanese and Chinese word segmentation. Previous work has attempted to detect and identify unknown words separately or jointly with word segmentation based on likelihood statistics [29, 82, 100, 137], character-level tagging [3, 90], and statistical rules or scores with linguistic knowledge, such as various Chinese morphological rules [62], Chinese monosyllabic words and personal names [11], Japanese character type heuristics [4, 57],[7] Japanese affixes [88], and Japanese orthographic variations [89].

In our work, we use a common baseline segmenter, introduced in §2.3, that can recognize any out-of-training-vocabulary (OOTV) words in principle because of the task formulation as character-level tagging, similar to previous methods introducing character-level unknown word processing [3, 90]. For better handling of OOTV words, our proposed methods can access additional lexical information: auto-segmented words in large unlabeled text in the method in Chapter 3, and source and target domain lexicons in the method in Chapter 4. Specifically, the latter method focuses on recognizing domain-specific words occurring in lexicons

[7] Unknown word processing in MeCab [57] was described in `https://taku910.github.io/mecab/`.

rather than on unknown words. As a result, the models have poor abilities to recognize unknown words in neither training words nor additional word vocabularies, as discussed in §3.4.4 and §4.4.6. We leave this problem for future work.

2.1.5. Related Work

For both Chinese and Japanese, word segmentation has been traditionally addressed by applying statistical learning algorithms, such as Markov models [2, 131, 162], maximum entropy [137, 153], conditional random fields (CRFs) [57, 100, 170], and logistic regression [92].

To reduce the burden of manual feature engineering for Chinese word segmentation, various neural network architectures have been explored. Specifically, character-based neural models have been developed to model the task as a sequence labeling problem, beginning with Zheng et al. [172] and Mansur et al.'s [75] earlier work that applied feed-forward neural networks. Furthermore, Pei et al. [99] used a neural tensor network to capture interactions between tags and characters. More sophisticated architectures have also been used as standard components of word segmentation models to derive effective features automatically. For instance, Chen et al. [13] proposed gated recursive neural networks (GRNNs) to model complicated combinations of characters, and Chen et al. [14] used long short-term memory (LSTM) [43] to capture long distance dependencies. Xu and Sun [150] combined LSTM and GRNNs to capture long term information better by utilizing chain and tree structures. Additionally, convolutional neural networks (CNNs) [20, 60] have also been used to extract complex features such as character n-grams [12, 142] and Chinese characters' graphical features [116]. Ma et al. [70] showed that a standard bidirectional LSTM (BiLSTM) [33] model can achieve state-of-the-art results when combined with deep learning best practices. Finally, Gan and Zhang [31] showed that a self-attention network (SAN) [140] achieved competitive results with BiLSTM and demonstrated further improvements by integrating a state-of-the-art model of contextualized representations BERT [25].

Word-based neural models have also been proposed. Typical word-based models [8, 9, 156, 166] sequentially determine whether or not to segment each character on the basis of word-level features and segmentation history, while retaining mul-

j	1	2	3	4
w_j	日本 (Japan)	語 (language)	まぢ (really)	ムズカシー (difficult)
$f_j : l_j$	1:2	3:3	4:5	6:10
p_j	名詞 (noun)	名詞 (noun)	副詞 (adverb)	形容詞 (adjective)
S_j	$\emptyset$	$\emptyset$	{まじ,マジ}	{難しい,むずかしい}

Table 2.3. Words in and labels of a sentence $\boldsymbol{x}$ =“日本語まぢムズカシー” (*nihon go maji muzukashī*), which means ‘Japanese language is really difficult.’

tiple segmentation candidates by beam search decoding. Liu et al. [67] combined neural architectures for segment (i.e., word) representations into a semi-CRF framework that searches for an optimal segmentation sequence of variable length segments. Using a deep CNN consisting of more than ten layers, Sun et al. [128] proposed a gap-based model to predict whether or not to segment two consecutive characters.

There has been less work applying neural models on Japanese word segmentation than for Chinese. Morita et al. [85] integrated an recurrent neural network (RNN) language model (LM) into a Japanese lattice-based morphological analysis (MA) framework [2, 4, 57, 131], which simultaneously predicts sequences of words and features, such as POS and lemma, over a word lattice of an input sentence. Kitagawa and Komachi [53] applied a pure neural model based on LSTM. Tolmachev et al. [133] demonstrated that a BiLSTM or 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.

2.2. Lexical Normalization

2.2.1. Task Definition

Lexical normalization is the task of identifying nonstandard words in a sentence and converting them into standard forms. We define a joint task of word Segmentation, POS tagging, and lexical Normalization (SPN) as follows: as shown in Table 2.3, a training instance for the SPN task is defined as a pair, comprising a sentence $\boldsymbol{x} = \boldsymbol{x}_{1:n} = (x_1, \ldots, x_n)$ and its label sequence $\boldsymbol{t} = \{(f_j, l_j, p_j, S_j)\}_{j=1}^{m}$, where n and m ($\leq n$) are the numbers of characters and words in $\boldsymbol{x}$, f_j and l_j are the indexes of the first and last character in j-th word w_j, and p_j is the POS tag of w_j. The set of standard forms S_j is equal to the empty set $\emptyset$ when w_j is a standard form, whereas S_j consists of one or more standard forms when w_j is a nonstandard form. Notably, our task formulation that assumes multiple standard forms for each nonstandard word is similar to that by Kaji and Kitsuregawa [50]. An SPN system is required to predict the word boundaries of an input sentence and the POS tag of each word, detect nonstandard words, and generate one of the standard forms of each nonstandard word.

2.2.2. Related Work

Word Segmentation and Lexical Normalization. For word segmentation and lexical normalization of Japanese UGT, most previous work applied the lattice-based MA framework. Nakamoto et al. [91] introduced an alignment method based on string similarity between original and variant forms into a minimum connectivity-cost MA method for chat text. Ikeda et al. [46] automatically constructed normalization rules of peculiar expressions in blogs, based on frequency, edit distance, and estimated accuracy improvements. Sasano et al. [112] defined derivation rules to recognize unknown onomatopoeia and variant forms of known words that frequently occur in webpages; their rules were also implemented in a recent MA toolkit Juman++ [134] to handle unknown words. Kaji and Kitsuregawa [51] developed a discriminative lattice traversal method for joint MA and normalization using hand-crafted rules similar to Sasano et al. [112]. Saito et al. [108] estimated character-level alignment from manually annotated pairs of formal and informal words on Twitter. Saito et al. [107] extracted formal-informal word pairs from unlabeled Twitter data based on semantic and phonetic similarity. In contrast to those methods, Ikeda et al. [47] applied a sequence-to-sequence model trained on synthetic formal-informal sentence pairs to sentence-level Japanese text normalization.

For Chinese, also an unsegmented language, nonstandard word detection and normalization methods have been proposed. Li and Yarowsky [61] extracted formal-informal word pairs using web-searched sentences defining informal words and a conditional log-linear ranking model. Wang and Ng [143] proposed beam-search decoding methods for lexical normalization as well as punctuation correction and recovery of missing words, for Chinese and English UGT, as preprocessing steps for Chinese↔English MT. Qian et al. [101] proposed a perceptron-based transition method with append, separate, and separate_and_substitute operations for the joint SPN task on Chinese microblog text. Zhang et al. [164] proposed a transition method using character-level and word-level LSTMs for word segmentation and detection of informal words.

English Lexical Normalization. Early work on lexical normalization of English SMS and microblog text employed a noisy channel formulation; to restore plausible standard forms from observed nonstandard words, Aw et al. [5] trained a statistical MT model and Choudhury et al. [17] trained a hidden Markov model on parallel sentences of standard and nonstandard English. Liu et al. [64] automatically collected training word pairs using carefully-designed web search queries and trained CRFs to calculate the conditional probability of a nonstandard character given a standard character. van der Goot [138] developed a state-of-the-art lexical normalization system for English and other European languages. The system called MoNoise is a random forest classifier with features including spell checker outputs, word embeddings, and n-gram probabilities. Muller et al. [87] adapted BERT for lexical normalization by introducing a subword alignment algorithm between standard and nonstandard words and a task-specific fine-tuning strategy.

Some other work has adopted unsupervised methods: a log-linear model to score standard and nonstandard word sequences [159], a graph-based method to model contextual and lexical similarity [119], and a finite-state transducer using word embedding and string similarity [104].

2.3. Baseline Model: BiLSTM

For a baseline model of our approaches in Chapters 3, 4, and 6, we use a BiLSTM architecture [33] that has been successfully applied to sequence labeling tasks

[16, 45]. The model consists of a character embedding layer, recurrent layers, and an inference layer.

Character Embedding Layer. Let $\mathcal{V}_c$ be a character vocabulary. Each character x_i in a given sentence $\boldsymbol{x}$ is transformed into a character embedding $\boldsymbol{e}^c_i$ of a d_c-dimensional vector by a lookup operation that retrieves the corresponding column of the embedding matrix $E_c \in \mathbb{R}^{d_c \times |\mathcal{V}_c|}$.

Recurrent Layers. An LSTM network addresses the issue of learning long-term dependencies and the gradient vanishing problem; we adopt a multi-layer and bidirectional variant of LSTM, i.e., BiLSTM. A sequence of character embeddings $\boldsymbol{e}^c_{1:n} = (\boldsymbol{e}^c_1, \cdots, \boldsymbol{e}^c_n)$ is fed into BiLSTM layers to derive contextualized representations $\boldsymbol{h}_{1:n} = (\boldsymbol{h}_1, \cdots, \boldsymbol{h}_n)$, which we call *character context vectors*. An l-th BiLSTM layer concatenates a forward hidden vector $\overrightarrow{\boldsymbol{h}}^{(l)}_i \in \mathbb{R}^{d_r}$ and a backward hidden vector $\overleftarrow{\boldsymbol{h}}^{(l)}_i \in \mathbb{R}^{d_r}$, which are calculated by forward LSTM (LSTM$_f$) and backward LSTM (LSTM$_b$). The outputs is a hidden vector $\boldsymbol{h}^{(l)}_i \in \mathbb{R}^{2d_r}$ for each time step i:

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

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

More concretely, each forward (backward) LSTM calculates forward (backward) hidden vectors $\overrightarrow{\boldsymbol{h}}_{1:n}$ ($\overleftarrow{\boldsymbol{h}}_{1:n}$) from an input sequence $\boldsymbol{v}_{1:n} = (\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n)$ of d_v-dimensional vectors as follows:

$$\begin{aligned}
\overrightarrow{\boldsymbol{h}}_i &= \mathrm{LSTM}_f(\boldsymbol{v}_i, \boldsymbol{h}_{i-1}) := \boldsymbol{o}_i \odot \mathrm{t}anh(\boldsymbol{c}_i) \ , \\
\boldsymbol{c}_i &= \boldsymbol{i}_i \odot \boldsymbol{t}_i + \boldsymbol{f}_i \odot \boldsymbol{c}_{i-1}, \\
\boldsymbol{o}_i &= \sigma(W_o \boldsymbol{v}_i + U_o \boldsymbol{h}_{i-1} + \boldsymbol{b}_o) \ , \\
\boldsymbol{i}_i &= \sigma(W_i \boldsymbol{v}_i + U_i \boldsymbol{h}_{i-1} + \boldsymbol{b}_i) \ , \\
\boldsymbol{f}_i &= \sigma(W_f \boldsymbol{v}_i + U_f \boldsymbol{h}_{i-1} + \boldsymbol{b}_f) \ , \\
\boldsymbol{t}_i &= \mathrm{t}anh(W_t \boldsymbol{v}_i + U_t \boldsymbol{h}_{i-1} + \boldsymbol{b}_t) \ ,
\end{aligned}$$

where $\odot$ denotes element-wise multiplication, σ is the sigmoid function, $\boldsymbol{i}$, $\boldsymbol{f}$, and $\boldsymbol{o}$ indicate an input gate, a forget gate, and an output gate, and W_i, W_o, W_f, W_t, U_i, U_o, U_f, $U_t \in \mathbb{R}^{d_r \times d_v}$ and $\boldsymbol{b}_i$, $\boldsymbol{b}_o$, $\boldsymbol{b}_f$, $\boldsymbol{b}_t \in \mathbb{R}^{d_r}$ are trainable parameters. Note that the input vector $\boldsymbol{v}_i$ (and its dimension d_v) corresponds to $\boldsymbol{h}_i^{(0)} = \boldsymbol{e}_i^c$ (d_c) for the first layer and to $\boldsymbol{h}_i^{(l-1)}$ ($2d_r$) for the second and subsequent layers. The backward LSTM similarly calculate backward hidden vectors $\overleftarrow{\boldsymbol{h}}_{1:n}$.

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

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

where $W_s \in \mathbb{R}^{|\mathcal{T}| \times 2d_r}$ and $\boldsymbol{b}_s \in \mathbb{R}^{|\mathcal{T}|}$ are trainable parameters. A label sequence is then output using a softmax or linear-chain CRF [59] layer. The softmax layer outputs a predicted label sequence $\boldsymbol{y} = y_{1:n} \in \mathcal{T}^n$ for $\boldsymbol{x}$ by selecting the elements of the score vector $\boldsymbol{s}_i$ with the largest value. The score of $\boldsymbol{y}$ is calculated as follows:

$$\text{Score}_{\text{softmax}}(\boldsymbol{x}, \boldsymbol{y}; \theta) = \sum_{i=1}^{n} s_{i,y_i} \ ,$$

where θ denotes all parameters of the model and s_{i,y_i} indicates the element of the score vector $\boldsymbol{s}$ corresponding to the i-th label y_i. By contrast, the CRF layer has a transition matrix $A \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{T}|}$ to provide transition scores between adjacent labels. The score of $\boldsymbol{y}$ is calculated as follows:

$$\text{Score}_{\text{CRF}}(\boldsymbol{x}, \boldsymbol{y}; \theta) = \sum_{i=1}^{n} (A_{y_{i-1}, y_i} + s_{i,y_i}).$$

We can find the best label sequence $y^\star$ by maximizing the sentence score as follows:

$$y^\star = \text{argmax}_{y \in \mathcal{T}^n} \ \text{Score}_{\text{CRF}}(\boldsymbol{x}, \boldsymbol{y}; \theta) \ . \tag{2.1}$$

An advantage of the softmax function is efficiency; the time complexity of standard training and inference for the softmax function is linear in the number of labels, whereas that for the linear chain CRF is quadratic. However, the CRF might achieve better accuracy for tasks where adjacent tokens are dependent, such as word segmentation. We use either the inference layer in each model in Chapter 3, 4, and 6, considering both the efficiency and accuracy.

Training Objective. During training, the parameters of the model are learned by minimizing the loss function L. The loss function for the softmax layer is defined as the cross entropy between gold and predicted label distributions over all sentences in training data $\mathcal{D}$:

$$L_{\text{softmax}}(\mathcal{D}) = -\sum_{(\boldsymbol{x},\boldsymbol{t})\in\mathcal{D}} \sum_{i} t_i \log \frac{exp(s_{i,t_i})}{\sum_{t_j\in\mathcal{T}} \exp(s_{i,t_j})} \ . \tag{2.2}$$

The loss function for the CRF layer is the negative log likelihood over all sentences in training data $\mathcal{D}$:

$$L_{\text{CRF}}(\mathcal{D}) = -\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \log p(\boldsymbol{y}|\boldsymbol{x},\theta) \ , \tag{2.3}$$

$$p(\boldsymbol{y}|\boldsymbol{x},\theta) = \frac{\text{score}(\boldsymbol{x},\boldsymbol{y};\theta)}{\sum_{y'} \text{score}(\boldsymbol{x},\boldsymbol{y'};\theta)} \ . \tag{2.4}$$

Notably, the Viterbi algorithm can be used for efficient calculation of the probability of a label sequence in Eq. (2.4), similarly to decoding in Eq. (2.1).

3. Character-to-Word Attention for Word Segmentation

3.1. Introduction

In recent years, neural network models have been widely applied to word segmentation, especially for Chinese, because they can minimize effort in feature engineering. Character-based models [13, 75, 99, 172] efficiently predict optimal label sequences by treating the task as sequence labeling that assumes first-order dependency between labels. Word-based models [8, 9, 156, 166] directly segment a character sequence into words and can easily achieve the benefits of word-level information. These neural models have achieved great success in Chinese word segmentation performance.

Within a sentence, a character has multiple candidate words that contain the character, but the plausibility of a candidate word differs within the target character's context. For example, more than three candidate words exist for characters x_3 =日 and x_4 =本 in the sentence $\boldsymbol{x}$ in Figure 3.1, so the proper word w_6 =日本 must be identified from among the candidates.[8] A feasible solution to develop a model with both characteristics is to incorporate word information into a character-based framework. Motivated by that consideration, we propose a character-based word segmentation model that incorporates word information into a BiLSTM-based architecture [14, 45]. Different from similar work in Chinese word segmentation [142, 157], we apply an attention mechanism [6, 69] to learn and distinguish the importance of all possible candidate words for a character within a context.

The contributions of this work are as follows:

[8] w_8 =日本人 can be correct depending on a segmentation criterion.

	j	1	2	3	4	5	6	7	8	No. of
		彼	は	日	本	人	日本	本人	日本人	cand.
i	$x_i \diagdown w_j$	*kare*	*wa*	*hi*	*hon*	*hito*	*nihon*	*honnin*	*nihonjin*	words
		(he)	(NOM)	(day)	(book)	(person)	(Japan)	(the person)	(Japanese)	
1	彼	1	0	0	0	0	0	0	0	1
2	は	0	1	0	0	0	0	0	0	1
3	日	0	0	1	0	0	1	0	1	3
4	本	0	0	0	1	0	1	1	1	4
5	人	0	0	0	0	1	0	1	1	3

Figure 3.1. An example of candidate words ($w_1, \ldots, w_8$) retrieved from a vocabulary for a sentence $\boldsymbol{x}$ =“彼は日本人,” which means ‘He is a Japanese.’ The value in each (i, j) represents whether the i-th character is included in the j-th word (i.e., δ_{ij} in Eq. 3.2).

- We introduced word information and an attention mechanism into a character-based word segmentation framework, to distinguish and leverage the importance of candidate words in different contexts.

- We empirically demonstrated that learning accurate attention to proper candidate words leaded to correct segmentation.

- Our model achieved better or competitive performance on both Japanese and Chinese datasets, compared with state-of-the-art word segmentation models.

3.2. Proposed Model

To disambiguate word boundaries more effectively than previous models, we integrate word information into the character-based framework, as shown in Figure 3.2. More specifically, we transform embeddings of multiple candidate words for each character into a fixed-size word vector, namely, a *word summary vector*, by a *word feature composition function.* In addition to the layers in the baseline BiLSTM-CRF model in §2.3, the proposed model comprises a word embedding layer, a word feature composition function, and additional recurrent layers.

The proposed model uses information on characters and candidate words in a given word vocabulary; that is, information on OOV words is unavailable if the characters of interest compose such words. For the word categories mentioned in §2.1.4, the proposed model does not explicitly distinguish whether each candidate

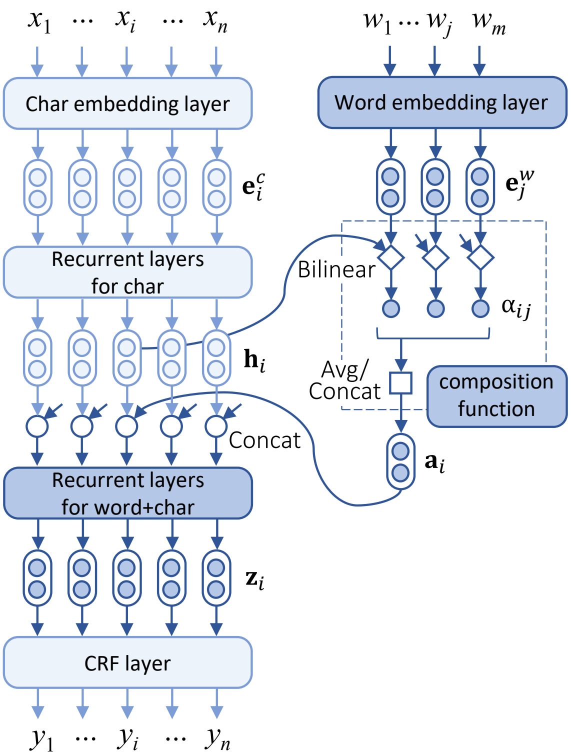


Figure 3.2. Architecture of the proposed model, which comprises the common components to the baseline model (light blue) and additional components (dark blue).

word is a simple, derivative, compound, monosemic, polysemic, or homographic word.

3.2.1. Word Embedding Layer

Given a character sequence $\boldsymbol{x} = \boldsymbol{x}_{1:n}$, we search for all words within a maximum word length corresponding to subsequences of the input sequence from a word vocabulary $\mathcal{V}_w$. We then obtain a set $\mathcal{W}_x = \{w_1, \cdots, w_m\}$ of all candidate words. For example, candidate words $\{w_1, \cdots, w_8\} = \{$彼$, \cdots,$日本人$\}$ are found for the sentence $\boldsymbol{x}$ in Figure 3.1. The embedding matrix $E_w \in \mathbb{R}^{d_w \times |V_w|}$ transforms each word $w \in \mathcal{W}_x \subseteq \mathcal{V}_w$ into a d_w-dimensional vector $\boldsymbol{e}^w$.

3.2.2. Composition Functions of Word Features

For each character x_i, a composition function aggregates embeddings of all candidate words that contain the character into a word summary vector $\boldsymbol{a}_i$. We introduce two attention-based composition functions, weighted average (WAVG) and weighted concatenation (WCON), that enable a model to pay more or less attention according to the importance of candidate words.

Both functions calculate the importance score u_{ij} of character x_i for word w_j in $\mathcal{W}_x$ through a bilinear transformation to capture the interaction between the character context vector $\boldsymbol{h}_i$ and the word embedding $\boldsymbol{e}^w_j$. Then to normalize scores, a softmax operation obtains the weight $\alpha_{ij} \in [0,1]$:

$$\begin{aligned} u_{ij} &= \boldsymbol{h}_i^T W_a \boldsymbol{e}^w_j \ , && (3.1) \\ \alpha_{ij} &= \frac{\delta_{ij} \exp(u_{ij})}{\sum_{k=1}^m \delta_{ik} \exp(u_{ik})} \ , && (3.2) \end{aligned}$$

where $W_a \in \mathbb{R}^{2d_r \times d_w}$ is a trainable parameter. To simplify equations, we introduce an indicator variable $\delta_{ij} \in \{0,1\}$ that is equal to 1 when the character x_i is included in the word w_j (Figure 3.1).

Next, WAVG and WCON calculate a word summary vector $\boldsymbol{a}_i$ as the weighted average and the weighted concatenation of word embeddings, respectively:

$$\boldsymbol{a}_i = \mathrm{WAVG}(x_i, \{w_j\}_{j=1}^m) = \sum_{j=1}^m \alpha_{ij} \boldsymbol{e}^w_j \ , \tag{3.3}$$

$$\boldsymbol{a}_i = \mathrm{WCON}(x_i, \{w_j\}_{j=1}^m) = \bigoplus_{l=1}^L \alpha_{i,i_l} \boldsymbol{e}^w_{i_l} \ , \tag{3.4}$$

where $\{w_j\}_{j=1}^m = \mathcal{W}_x$ and $\bigoplus(\cdot)$ indicates the concatenation of given arguments. Let K be the maximum word length, $L = \sum_{k=1}^K k$, and i_l for the character x_i denotes the corresponding index in $\mathcal{W}_x$ of l-th words w'_l in the list $\mathcal{W}'_{x,i} = \{w'_1, \cdots, w'_L\} = \bigcup_{k=1}^K \bigcup_{p=-(k-1)}^0 \{x_{i+p:i+p+k-1}\}$. If $w'_l \notin \mathcal{V}_w$, we use a zero vector as the l-th argument in Eq. (3.4). For example, if $K=3$, WCON concatenates embeddings of words corresponding to x_i (length 1), $x_{i-1:i}$, $x_{i:i+1}$ (length 2), $x_{i-2:i}$, $x_{i-1:i+1}$ and $x_{i:i+2}$ (length 3) in that order, into a single vector for the character x_i. WAVG and WCON finally output a summary vector of size d_w and Ld_w, respectively. The obtained summary vector $\boldsymbol{a}_i$ and the context vector $\boldsymbol{h}_i$ are together fed into the subsequent layer. We use a zero vector as a summary vector if no candidate words for a character are found. We describe a calculation example of WAVG and WCON-based word summary vectors in Appendix §A.

We also use two more variants of composition functions without the attention mechanism, the average function (AVG) and the concatenation function (CON). AVG is a special case of WAVG, where $\alpha_{ij} = \delta_{ij} / \sum_k \delta_{ik}$ for all (i, j) in Eq. (3.3). CON is the equivalent function to the word features used in Wang and Xu's work [142] and a special case of WCON, where $\alpha_{i,i_l} = 1$ for all (i, i_l) in Eq. (3.4).

Notably, our importance score function in Eq. (3.1) has the same form as the bilinear variant of the global attention model by Luong et al. [69], which was used to calculate alignments between source and target hidden vectors in MT. They further evaluate the input-feeding approach that uses previous alignment information in next-time steps. To take into account attended words from previous characters, similar word segmentation approaches might also be useful. We leave this for future work.

3.2.3. Recurrent Layers for Word-Integrated Character Representation

The summary vector $\boldsymbol{a}_i$ and the context vector $\boldsymbol{h}_i$ for a character are together fed into additional recurrent layers, which are BiLSTM layers, to further contextualize character representations using word information of surrounding characters. Given the input $\boldsymbol{h}_i \oplus \boldsymbol{a}_i$, hidden vectors are calculated, and the hidden vectors $\boldsymbol{z}_{1:n}$ of the last BiLSTM layer are fed into the CRF layer.

3.3. Experimental Settings

Datasets. We evaluated our model on five Chinese and Japanese datasets. For in-domain word segmentation, we used the Chinese Penn Treebank 6.0 (CTB6)[9] [152], MSR from the second International Chinese Word Segmentation Bakeoff[10] [28], and the Balanced Corpus of Contemporary Written Japanese (BCCWJ)[11] version 1.1 [72] with the short unit word (SUW).

[9] https://catalog.ldc.upenn.edu/LDC2007T36

[10] http://sighan.cs.uchicago.edu/bakeoff2005/

[11] https://pj.ninjal.ac.jp/corpus_center/bccwj/en/

	Chinese		Japanese			
	CTB6	MSR	BCCWJ	JDC	JMC	
Domain	News	News	News+	News+	News	Web
Train size	23K	83K	56K	30K	48K	
Dev size	2.0K	4.3K	1.0K	0.5K	0.5K	0.5K
Test size	2.9K	4.0K	3.0K	2.0K	2.0K	2.0K

(i) Source domain data

	Japanese						
	JDC			JMC			
Domain	Journal	Patent	Recipe	Sports	Phone	Dining	Travel
Test size	0.3K	2.0K	0.7K	0.5K	1.3K	0.9K	1.5K

(ii) Target domain data

Table 3.1. The size (the number of sentences) of each dataset.

For cross-domain word segmentation, we used the Japanese Dependency Corpus (JDC)[12] [83] and a Japanese mixed corpus, which we call JUMAN Mixed Corpus (JMC). JDC follows the criterion that extends the SUW by separating inflectional words' endings from their stems, and consists of six domains: BCCWJ (parts of sentences in BCCWJ), economy newspaper articles, dictionary example sentences, information processing journal abstracts, patent specifications, and recipe text. We used sentences from the former three domains as source domain data and sentences from other domains, i.e., journal, patent, or recipe, as target domain data. JMC comprises three corpora that follow the same criterion for the Japanese morphological analyzer JUMAN: Kyoto University Text Corpus (KTC)[13] Version 4.0 [58], Kyoto University Web Document Leads Corpus (KWDLC)[14] [36], and Kyoto University and NTT Blog Corpus (KNBC)[15] [37]. We used news sentences in KTC and Web sentences in KWDLC as source domain data and blog text with a specific topic (sports, mobile phones, dining, or travel in Kyoto) in KNBC as target domain data.

The statistics of the datasets are shown in Table 3.1. We followed the same

[12] http://www.lsta.media.kyoto-u.ac.jp/resource/data/word-dep/home-e.html

[13] http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?Kyoto%20University%20Text%20Corpus

[14] http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KWDLC

[15] http://nlp.ist.i.kyoto-u.ac.jp/kuntt/KNBC_v1.0_090925.tar.bz2

training/development/test split as in previous work [14, 158] for CTB6, official training/test split for MSR. For BCCWJ, we used the "ClassA-1" sentences defined for the Japanese Dependency Corpus (JDC)[16] as test data and the remining sentences in the BCCWJ core data as training data. We randomly selected 90% of the sentences in the training data as a training set and used the other 10% as a development set, respectively for CTB6 and MSR. Also, we randomly selected a certain number (500, 1000, or 2000, as in Table 3.1) of sentences for the development set of BCCWJ, and for the respective development and test sets of the JDC and JMC source domain data.

Word Vocabulary Construction. Apart from given training and development sets for each dataset, we assumed that no annotated information, including external dictionaries and third-party segmenters, was available in our experiments. Therefore, we used the training set and large unlabeled text to obtain a word vocabulary for our proposed model.

First, we trained a baseline model from each training set and applied it to unlabeled text. Then, we treated the union of auto-segmented words from the text and gold words from the training set as a word vocabulary. From the auto-segmented text, we discarded words occurring less than the minimum word frequency threshold f of five, which is the default value in Word2Vec that we used for pre-training word embeddings, as described later in this subsection. We used approximately 5.9M sentences in the non-core data of BCCWJ (BCCWJ-NC)[17] for the Japanese datasets and 48M sentences in Chinese Gigaword Fifth Edition[18] for the Chinese datasets as unlabeled text.

In the implementation of our proposed model, a word dictionary manages all words in a given word vocabulary. In the training phase, however, the model holds embedding parameters for only a part of those words corresponding to gold words or substrings in training sentences. In the test phase, the model searches for substrings in the test sentences from the dictionary and dynamically loads their

[16] JDC is available at `http://www.lsta.media.kyoto-u.ac.jp/resource/data/word-dep/`. We list the document IDs in the ClassA-1 set in Appendix §C.

[17] We restored provided auto-segmented text, which were segmented by another segmenter, to the original raw sentences and used them as unlabeled text.

[18] `https://catalog.ldc.upenn.edu/ldc2011t13`

Method	Parameter	Value
Baseline/Proposed model	Character embedding size (d_c)	300
	Number of BiLSTM-C layers	2
	Number of BiLSTM-C 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 model	Word embedding size (d_w)	300
	Number of BiLSTM-WC layers	0 or 1
	Number of BiLSTM-WC hidden units	0 or 600
	Word vector dropout rate	0.4
	Minimum word frequency (f)	5
	Maximum word length (K)	4

Table 3.2. Hyperparameter values common between the baseline and the proposed model (top) and specific to the proposed model (bottom). BiLSTM-C and BiLSTM-WC, respectively, indicate recurrent layers for character and word-integrated character representations.

word embeddings from the external word embedding model used for initialization. This strategy reduces model size while handling hundreds of thousands of words in the dictionary, as shown later in §3.4.3.

Pre-training of Embedding Parameters. Following previous work [21], we pre-trained word embeddings from large text and used them to initialize the word embedding matrix in our proposed segmenter. To pre-train word embeddings, we applied the gensim [105] implementation of Word2Vec [77] to the same text as those used to construct word vocabularies, i.e., auto-segmented sentences in BCCWJ-NC or Chinese Gigaword processed by the baseline segmenter. We used the toolkit with the skip-gram algorithm, embedding size 300, the number of iterations one, and other default parameters, including minimum frequency five. For words occurring only in a training set, we randomly initialized their embeddings. We fine-tuned all word embeddings during training of the proposed segmenter.

In contrast, we randomly initialized all character embeddings, since pre-trained character embeddings did not improve performance in our preliminary experiments.

	Chinese		Japanese			
Method	CTB6	MSR	BCCWJ	JDC	JMC	
	News	News	News+	News+	News	Web
BASE	95.44	97.47	99.06	98.29	98.74	96.90
AVG	95.90†	98.50†	99.19†	98.31	98.73	97.13
WAVG	95.96†	**98.55**†‡	99.24†	98.42†‡	98.80	97.26†
CON	96.06†	98.52†	99.38†	98.50†	**98.92**†	97.48†
WCON	**96.29**†‡	98.52†	**99.43**†	**98.54**†	**98.91**†	**97.63**†‡

Table 3.3. In-domain performance on the development sets. The table shows the means of F_1 scores of the baseline (BASE) and proposed model variants. The symbols † and ‡ indicate statistical significance at the 0.01 level over the baseline and over the variant without attention, respectively.

Hyperparameter Setting. Table 3.2 shows the proposed model's hyperparameters. We introduced additional one-layer BiLSTM for word-integrated character representation (BiLSTM-WC) only for Chinese datasets, according to our preliminary experiments. We set the maximum word length K to 4 because this value covered 99% of words in the most development sets and larger values did not further improve performance in our preliminary experiments.[19] The dropout strategy [161] was applied to non-recurrent connections of recurrent layers. Besides that, we used word vector dropout, which randomly replaces a word embedding to a zero vector when calculating a word summary vector in Eq. (3.3) or Eq. (3.4). We used a mini-batch stochastic gradient descent to optimize parameters and reduced the learning rate with a fixed decay rate every epoch after the first five epochs; we trained models for up to 20 epochs and selected the model with the highest F_1 score on the development set.

3.4. Results and Analysis

3.4.1. Main Results

Comparison of Proposed Model Variants. We evaluated our baseline and proposed model variants on the development sets of the five datasets, and Table

[19] However, an optimal value of K can vary depending on domain, as we show later in §3.4.3.

Method	Model size	BCCWJ	JDC	JMC	
		News+	News+	News	Web
BASE	13.9M	99.06	98.29	98.74	96.90
BASE_L	32.8M	-0.03	+0.04	-0.13	+0.08
WAVG	29.5M	+0.18	+0.13	+0.06	+0.36
WAVG_L	32.7M	+0.17	+0.07	+0.08	+0.39
CON	29.1M	+0.32	+0.21	+0.18	+0.59
CON_L	32.6M	+0.29	+0.15	+0.17	+0.65
WCON	32.7M	+0.37	+0.24	+0.18	+0.73

Table 3.4. In-domain performance on the Japanese development sets and size of model variants with original and larger size (denoted as "_L"). Differences between BASE and each model's mean F_1 scores are shown.

3.3 displays F_1 scores of each model variant. We report the mean and standard deviation of F_1 scores of three runs for each model, unless otherwise specified. Among the proposed model variants, WCON achieved the best performance in almost all cases. From the results, we observed the following three findings. First, the four word-integrated model variants outperformed the pure character-based baseline on almost all datasets. Each variant's improvement over the baseline was significant at the 0.01 level in 20 of 24 cases, according to McNemar's tests [32] on differences between two systems' word-level predictions for gold words, i.e., true positive (TP) or false negative (FN). Second, the attention-based variants achieved performance equivalent to or better than their counterparts without attention. According to McNemar's tests, the improvements of WAVG over AVG on the MSR and JDC news set and those of WCON over CON on the CTB6 and JMC Web set were statistically significant. Third, the concat-based variants performed better than their average-based counterparts in almost all cases, probably because CON and WCON retain word length and character position information. For example, (d_w+1)-th to $2d_w$-th dimensions of a summary vector always represent a word with a length of two ending with a target character (namely $x_{i-1:i}$ for x_i), while AVG and WAVG lose such information.

Table 3.4 shows the size of model variants, i.e., the number of parameters trained with the Japanese datasets (see Appendix §B for more details). The proposed model variants have 109–135% more parameters than the baseline, and

Method	Chinese		Japanese			
	CTB6	MSR	BCCWJ	JDC	JMC	
	News	News	News+	News+	News	Web
BASE	95.40	96.67	98.70	98.09	98.52	96.96
WCON	**96.38**	97.79	**99.00**	**98.49**	**98.77**	**97.49**
Sun+ ’17 [128]	96.3	97.9	–	–	–	–
Wang+ ’17 [142]	–	**98.0**	–	–	–	
Zhou+ ’17 [173]	96.2	97.8	–	–	–	
Neubig+ ’11 [92]	–	–	98.32	98.06	98.35	96.92
Kitagawa+ ’18 [53]	–	–	98.42	98.07	98.12	97.17

(i) In-domain results

Method	Japanese								
	JDC				JMC				
	Journal	Patent	Recipe	Δ	Sports	Phone	Dining	Travel	Δ
BASE	97.38	94.51	93.83	-2.85	93.56	94.89	93.46	94.33	-3.68
WCON	**97.87**	**96.61**	**94.99**	-2.00	**94.56**	**95.68**	**94.34**	**95.12**	-3.20
Neubig+ ’11 [92]	97.09	95.05	93.88	-2.72	93.01	94.79	93.24	93.94	-3.89
Kitagawa+ ’18 [53]	97.55	92.99	93.97	-3.23	93.74	95.09	93.91	94.15	-3.42

(ii) Cross-domain results

Table 3.5. In-domain and cross-domain performance on the test sets. Δ denotes the difference of the average score among target domains from that among source domains.

WCON has 11–12% more parameters than other variants. Then, for a fair comparison of model size, we examined the performance of a larger size of model variants, BASE_L, WAVG_L, and CON_L, with numbers of parameters similar to WCON, by increasing the number of BiLSTM hidden units d_r, which was set to 960, 675, and 680, respectively. Large model variants exhibited small performance improvements or degradation compared to variants with the original size, but none achieved better performance than WCON, indicating that the performance differences among model variants were due not to the different model size but to the different model structures. Note that the performance of the current size model might not be saturated in the Web domain, because all large model variants consistently show performance improvements.

Test Set Performance and Comparison with Existing Methods. We evaluated WCON, our best model variant, on the test sets and compared it with the baseline. Table 3.5 shows F_1 scores and OOV recall and Table 3.6 shows OOV

		JDC			JMC			
		Journal	Patent	Recipe	Sports	Phone	Dining	Travel
OOV rate	for $\mathcal{V}_{\text{train}}$	5.58	9.51	6.91	4.40	4.12	5.56	4.54
	for $\mathcal{V}_{\text{train}} \cup \mathcal{V}_{\text{auto}}$	2.06	2.14	1.23	1.74	1.57	1.93	1.52
OOV recall	BASE	87.01	82.51	76.61	64.01	65.90	65.84	66.94
	WCON	**87.62**	**86.21**	**79.21**	**68.22**	**69.41**	**68.27**	**67.99**

Table 3.6. Dataset statistics and performance of models for OOV words. The OOV rate of a test set for vocabulary $\mathcal{V}$ indicates the rate of words that occur in the test set but are not in $\mathcal{V}$. OOV recall indicates recall for words not in the training vocabulary $\mathcal{V}_{\text{train}}$.

rates of the datasets. WCON improved F_1 scores over the baseline by 0.25–1.0 points on source domains and approximately 0.5–2.1 points on target domains. As indicated by its smaller performance drops from source domain results (Δ values), WCON showed robust performance for domain shifts in cross-domain settings. Among target domains, the model obtained the largest gain in the patent domain. This can be explained in terms of OOV words shown in Table 3.6. Although in particular, this domain has many OOV words not in the training vocabulary $\mathcal{V}_{\text{train}}$, a large portion (77%) were covered by the auto-segmented word vocabulary $\mathcal{V}_{\text{auto}}$ (with $f = 5$ and $K = 4$), and the model also greatly improved OOV recall. Thus, our model exploited word information not covered by training data while also reducing substantive unknown words.

As shown in Table 3.5, we also compared our best model, WCON, with state-of-the-art models without additional annotated data. In the in-domain setting, we obtained better performance on BCCWJ, JDC, JMC, and CTB6 in comparison with the best previous models.[20] On MSR, we obtained comparable performance with the character-based model with word features by Wang and Xu [142], which used different unlabeled texts from ours to pre-train word embeddings. In the cross-domain setting, our model achieved better performance than two existing Japanese segmenters in all domains owing to effective information on candidate words, because neither used direct word information except for word indicator features.

[20] As mentioned in §3.5, several work achieved further better performance on the Chinese datasets at nearly the same time as and later than our original work.

3.4.2. Effect of Semi-Supervised Learning

Method	Use of unlabeled text	BCCWJ	JDC	JMC	
		News+	News+	News	Web
BASE		99.06	98.29	98.74	96.90
CON (random)		99.10	98.38	98.65	96.88
WCON (random)		99.06	98.28	98.66	97.20
BASE +self-training	✓	99.21	98.34	98.76	97.07
CON (pre-trained)	✓	99.38	98.50	**98.92**	97.48
WCON (pre-trained)	✓	**99.43**	**98.54**	98.91	**97.63**

Table 3.7. Performance on the development sets in pure-supervised and semi-supervised settings. "Random" and "pre-trained" indicate models started respectively with randomly initialized word embeddings and with pre-trained word embeddings.

Our proposed model is a semi-supervised learning method that uses unlabeled text for pre-training of word embedding parameters. To investigate the contribution of unlabeled text, we evaluated both a pure-supervised version of the proposed model, which began from randomly initialized word embedding parameters, and a semi-supervised version of the baseline model, which additionally used auto-segmented text through self-training. Notably, we use the same experimental settings as in §3.3.

As Table 3.7 shows, both supervised versions of proposed model variants CON and WCON, with and without attention, could obtain little performance improvement over the baseline and seriously underperformed their semi-supervised versions. This suggests that, from the performed task that predicts segmentation labels from character representations incorporated with word vectors, learning meaningful word representations is difficult. Rather than using an external method such as the skip-gram, another effective method might be to train a model from an auxiliary task, like word-level language modeling, along with the segmentation task. Either way, considering the sparse distribution of words, large amounts of text are probably necessary. From comparison between baseline and proposed models in the semi-supervised setting, we observed limited performance improvements through self-training, thus indicating that our proposed model more effectively utilizes unlabeled text.

f	BCCWJ		JDC		JMC	
	Dict	WE	Dict	WE	Dict	WE
1	591K	77K	634K	51K	794K	87K
5	174K	56K	169K	36K	214K	60K
10	122K	51K	113K	32K	143K	53K

(i) Vocabulary size for each dataset

	f	BCCWJ	JDC	JMC	
		News+	News+	News	Web
OOV rate	1	0.53	0.30	0.61	0.94
	5	0.71	0.38	0.73	1.15
	10	0.84	0.46	0.82	1.28
F_1	1	98.97	98.46	**98.80**	**97.54**
	5	**99.00**	**98.49**	98.77	97.49
	10	98.93	98.47	98.68	97.45

(ii) OOV rate and F_1 on the source domain test sets

	f	JDC			JMC			
		Journal	Patent	Recipe	Sports	Phone	Dining	Travel
OOV rate	1	1.73	1.52	1.00	1.49	1.47	1.58	1.25
	5	2.06	2.14	1.23	1.74	1.57	1.93	1.52
	10	2.24	3.28	1.83	1.90	1.76	2.09	1.72
F_1	1	98.06	**96.61**	**95.19**	**94.59**	**95.75**	**94.44**	**95.17**
	5	97.87	**96.61**	94.99	94.56	95.68	94.34	95.12
	10	**98.13**	96.39	94.97	94.51	95.71	94.13	95.10

(iii) OOV rate and F_1 on the target domain test sets

Table 3.8. Vocabulary size, OOV rate, and performance of the WCON model with each minimum word frequency threshold f on the test sets.

3.4.3. Effect of Word Frequency and Length Thresholds

We analyzed how the performance of the WCON model changed in various domains according to different word vocabularies in both minimum frequency and maximum length of words. For the minimum frequency threshold f, a model's vocabulary excludes words occurring fewer time than the threshold value in auto-segmented text. For the maximum length threshold K, a model ignores words whose length exceeds the threshold value. Namely, a smaller frequency threshold and a larger length threshold lead to a larger vocabulary.

Minimum Word Frequency Threshold. First, we fixed the length threshold to 4 and changed the word frequency threshold f among {1, 5, 10}. Then we

K	F$_1$ for each word length k								Vocab size		OOV
	1	2	3	4	5	6	≥ 7	All	Dict	WE	rate
	(66.39)	(27.24)	(3.57)	(1.87)	(0.59)	(0.19)	(0.16)	(100)			
0	98.67	97.94	93.08	93.61	92.91	90.13	87.07	98.09	–	–	1.78
1	98.69	97.98	93.06	93.85	92.21	90.37	89.90	98.12	27K	3K	1.66
2	98.81	98.31	93.59	93.66	93.22	90.31	89.87	98.32	107K	24K	0.83
3	98.89	98.32	94.01	94.28	93.44	90.75	88.72	98.39	141K	32K	0.57
4	**98.95**	98.39	94.30	95.23	94.19	91.55	89.64	98.49	169K	36K	0.44
5	98.90	**98.44**	**94.68**	**95.72**	**95.41**	**92.29**	**90.69**	**98.51**	183K	37K	0.40
6	98.91	98.39	94.47	95.62	94.12	91.67	90.46	98.48	189K	37K	0.40

(i) JDC news+ test set

K	F$_1$ for each word length k								Vocab size		OOV
	1	2	3	4	5	6	≥ 7	All	Dict	WE	rate
	(59.26)	(31.68)	(5.07)	(2.45)	(0.84)	(0.44)	(0.44)	(100)			
0	95.96	94.25	87.99	89.09	80.45	77.75	59.24	94.51	–	–	9.51
1	95.76	93.87	87.51	88.62	78.57	74.37	60.14	94.21	27K	3K	8.71
2	97.13	96.49	88.87	90.99	82.59	78.50	61.63	96.04	107K	24K	4.73
3	97.36	**96.87**	91.85	92.51	84.66	82.42	63.32	96.53	141K	32K	2.90
4	**97.37**	96.84	92.61	92.98	87.34	83.46	64.02	96.61	169K	36K	2.14
5	97.30	96.80	92.98	92.96	87.71	85.37	60.80	96.58	183K	37K	1.83
6	97.33	96.84	**93.43**	**93.84**	**88.50**	**86.69**	**65.96**	**96.68**	189K	37K	1.66

(ii) JDC patent test set

Table 3.9. Performance of the WCON model with maximum word length K and that of the baseline denoted with $K = 0$ for each word length k (JDC). Values in "()" denote percentages of words of length k in the data. We highlighted with gray background the results of the model with $K \geq k$ that outperformed the model with $K < k$ for each k.

examined vocabulary size, OOV rates, and the performance of WCON as shown in Table 3.8. The increase in the frequency threshold from 1 to 10 resulted in approximately an 80% smaller vocabulary, which corresponds to 35% smaller number of embedding parameters, but drops of 0.2–0.3 points for source domains and 0.3–1.8 points for target domains in OOV rates. The model with $f = 1$ achieved (close to) the best performance on all datasets. The model with the larger frequency threshold achieved similar performance on some source domains but yielded lower performance on most target domains with relatively higher OOV rates. This suggests that infrequent words in a source domain is still useful for segmenting out-of-domain text.

K	F$_1$ for each word length k								Vocab size		OOV
	1	2	3	4	5	6	≥ 7	All	Dict	WE	rate
	(47.20)	(35.14)	(10.07)	(5.04)	(1.55)	(0.57)	(0.42)	(100)			
0	98.08	97.53	94.43	94.14	89.21	83.58	68.22	96.96	–	–	3.11
1	98.10	97.47	94.73	94.28	90.09	83.58	70.20	97.01	51K	3K	3.01
2	98.31	97.85	94.89	94.11	89.51	85.11	69.66	97.25	126K	33K	2.11
3	98.42	98.07	95.55	94.73	90.45	85.48	68.90	97.49	172K	50K	1.57
4	98.42	98.08	95.49	95.08	90.08	84.12	68.41	97.49	214K	60K	1.15
5	98.40	98.07	95.77	95.31	91.47	86.34	**70.97**	97.57	238K	65K	0.91
6	**98.45**	**98.15**	**95.82**	**95.46**	**92.48**	**87.02**	70.89	**97.66**	250K	67K	0.83

(i) JMC Web test set

K	F$_1$ for each word length k								Vocab size		OOV
	1	2	3	4	5	6	≥ 7	All	Dict	WE	rate
	(48.66)	(29.40)	(14.35)	(5.63)	(1.35)	(0.38)	(0.23)	(100)			
0	94.77	94.04	92.08	89.35	81.04	72.79	36.77	93.45	–	–	5.56
1	94.61	93.90	92.05	89.46	80.98	70.85	40.43	93.33	51K	3K	5.34
2	94.93	94.54	92.59	89.50	81.14	70.71	**43.01**	93.76	126K	33K	4.13
3	95.01	94.90	92.91	89.90	81.42	72.14	40.90	93.98	172K	50K	2.94
4	95.17	95.19	93.51	91.22	82.31	**75.66**	40.27	94.34	214K	60K	1.93
5	95.22	95.23	**93.90**	**91.76**	83.57	75.60	40.17	**94.48**	238K	65K	1.61
6	**95.25**	**95.28**	93.64	91.65	**84.01**	74.53	38.75	94.46	250K	67K	1.47

(ii) JMC dining test set

Table 3.10. Performance of the WCON model with maximum word length K and that of the baseline denoted with $K = 0$ for each word length k (JMC).

Maximum Word Length Threshold. Second, we fixed the frequency threshold to 5, changed the word length threshold K among $\{1, 2, \cdots, 6\}$, and evaluated the performance for each length of gold words in evaluation sentences. We picked up several test sets from JDC and JMC data, which are source domain data and target domain data with higher OOV rates (Table 3.9–3.10). We also show the performance of the baseline as the model with "$K = 0$" for reference.

OOV rates for the model with the largest length threshold $K = 6$ decreased up to 7 points from the model with $K = 1$, and performance also varied greatly. For each length k of gold words, the model using words of the length, i.e., the model with $K \geq k$, tended to outperform the model not using those words, i.e., the model with $K < k$, as highlighted by gray background in Table 3.9–3.10. Moreover, the model with the larger threshold value often improved the performance for shorter words, and, therefore, overall performance. These results suggest that information on words of a particular length was effectively used to disambiguate character sequences of the same or shorter length.

For each data domain, performance was saturated when $K = 5$ for the news domain (the rate of words whose length $k \geq 6$ is 0.35%) and the dining domain (0.6%). In contrast, better performance was obtained when $K = 6$ for the patent domain (0.9%) and the Web domain (1%). Especially for domains with many long words, such as loanwords written with katakana, we can expect to achieve robust segmentation by a model with larger maximum word length.

3.4.4. Effect of Attention for Segmentation Performance

To analyze how the attention mechanism affects segmentation performance, we show segmentation accuracy and attention accuracy of WCON for the BCCWJ development set in Figure 3.3. Segmentation accuracy indicates character-level accuracy of segmentation label prediction. Attention accuracy is defined as the rate of characters that correctly attend to gold words.[21]

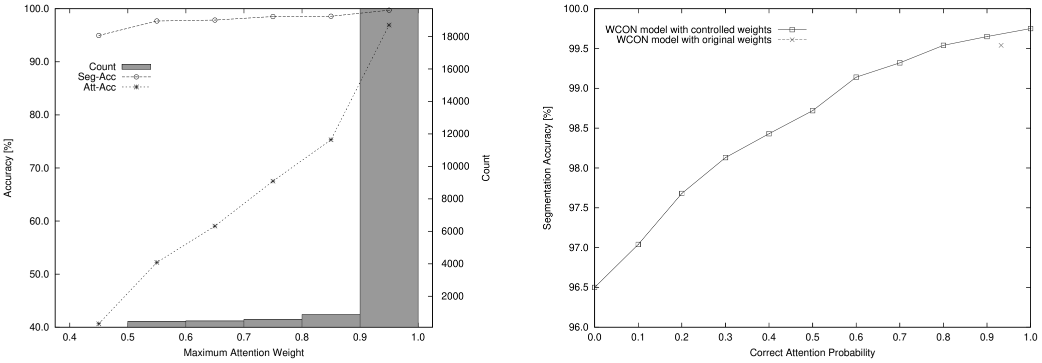
In Figure 3.3 (i), we show the count of corresponding characters and accuracy for each case of attention status: (a) there are no candidate words (then word vectors are not available); (b) there are candidate words except for a gold word (then attention is always incorrectly paid); (c) only a gold word is a candidate (then attention is always correctly paid); and (d) candidate words consist of both a gold and other words (then attention can be correctly paid if weights are properly calculated). Compared to overall accuracy, the model resulted in much poor segmentation accuracy when there were no gold words, that is, characters of interest composed OOV words (case b); However, it achieved better accuracy when gold words were available (cases c and d), with the benefit of proper word information. Case (c), in which almost perfect segmentation accuracy was achieved, might have the most easily identified correct labels due to less ambiguity. In case (d), the model successfully paid attention at a rate of more than 93% and achieved much higher segmentation accuracy than in case (b).

In Figure 3.3 (ii), we investigated detailed performance in case (d); we divided

[21] We regarded that a character attended to the word with the largest weight among all candidate words, and judged as correct if the attended word corresponded to the gold segmentation.

Possibility of correct attention	No. of candidates	Existence of gold word	Count	Seg-Acc	Att-Acc
(a) Non-available	0	0	0	–	–
(b) Always incorrect	≥1	0	1614	95.62	0.00
(c) Always correct	1	1	10561	99.96	100.00
(d) Otherwise	≥2	1	22222	99.54	93.25
Overall	≥0	0 or 1	34397	99.48	90.93

(i) Segmentation accuracy (Seg-Acc) and attention accuracy (Att-Acc) for each case of attention possibility



(ii) Segmentation accuracy and attention accuracy for each case of maximum attention weight $\alpha_{i,j^\star}$

(iii) Segmentation accuracy when attention weights were artificially controlled so that the correct attention probability p_t corresponded to a particular value in $\{0, \cdots, 1\}$

Figure 3.3. Effect of the WCON model's attention for segmentation performance on the BCCWJ development set.

all examples of characters into intervals from $[0, 0.1)$ to $[0.9, 1.0]$[22] on the basis of the maximum value of attention weights to (one of the) candidate words and evaluated both accuracy for each interval.[23] As Figure 3.3 illustrates, distribution of maximum weight $\alpha_{i,j^\star}$ was biased toward a higher value, that is, the case in which $\alpha_{i,j^\star} \geq 0.9$ corresponded to about 89% of all cases. Both attention

[22] We omitted intervals from $[0, 0.1)$ to $[0.3, 0.4)$ with infrequent examples from the figure. They had only 21 examples in total (0.1% of all), and the corresponding segmentation accuracy were close to 90%.

[23] For example, if there are two characters that one attends to its gold word with the weight of 0.95 and the other attends to an incorrect word with the weight of 0.95, then attention accuracy for $[0.9, 1.0]$ is 1/2.

and segmentation accuracy improved with increased value of $\alpha_{i,j^\star}$, and therefore, this confidence score properly reflected the model's certainty of prediction. We obtained high segmentation accuracy (99.7%) in the most confident case in which $\alpha_{i,j^\star} \geq 0.9$.

To examine whether a direct relationship exists between attention accuracy and segmentation accuracy, we controlled correctness of attention by artificially changing values of attention weights of the trained model and evaluated segmentation accuracy for each "correct attention probability." Specifically, on the basis of the correct attention probability threshold $p_t \in [0, 1]$, a random variable $p \sim \text{Uniform}(0, 1)$, and gold labels, we changed a weight value α_{ij} of the trained WCON model for a character x_i and a word w_j as follows:

$$\begin{cases} \alpha_{i,j \neq g} = \frac{1}{L} \quad \text{and} \quad \alpha_{i,g} = 1 - \frac{m-1}{L} & (p < p_t), \\ \alpha_{i,j \neq j_c} = \frac{1}{L} \quad \text{and} \quad \alpha_{i,j_c} = 1 - \frac{m-1}{L} & (\text{otherwise}), \end{cases}$$

where m denotes the number of candidate words $\{w_j\}_{j=1}^m$ for the character x_i, $L = \sum_{k=1}^K k = 10$ indicates the maximum number of candidate words, g indicates the index of the gold word, and j_c indicates the index of a randomly chosen candidate word except for the gold word. Namely, given the threshold value p_t, the model (forcibly) pays correct attention with the probability p_t, while assigning small weights to other candidate words. As in Figure 3.3 (iii), segmentation accuracy monotonically improved according to the increase of correct attention probability, indicating that our model tends to adopt candidate word information emphasized by attention weights for segmentation decisions and that learning accurate attention to proper words leads to correct segmentation. Possibly, therefore, overall segmentation performance can be further improved by learning more accurate attention or discarding words with low confidence.

3.4.5. Effect of Additional Word Embeddings from Target Domains

Aiming to improve cross-domain performance, we tried and evaluated a simple method to enhance our model with target domains' unlabeled text. Specifically, for each JDC and JMC data, we merged unlabeled text of source and all target

Dataset	Domain		Unlabeled text	No. of sent	No. of char
JDC/JMC	Source	Various	Text of various genres (BCCWJ-NC)	5.9M	193.3M
JDC	Target	Journal	Abstracts in computer science papers	1.0M	57.8M
		Patent	Patent publication	1.0M	63.3M
		Recipe	Recipe text	1.0M	38.0M
JMC	Target	Sports Phone Dining Travel	Posts with the corresponding categories of question answering sites	2.8M	94.6M

Table 3.11. Unlabeled text of source and target domains

Method	JDC				JMC					
	Source	Target			Source		Target			
	News+	Journal	Patent	Recipe	News	Web	Sports	Phone	Dining	Travel
BASE	98.09	97.38	94.51	93.83	98.52	96.96	93.56	94.89	93.46	94.33
+ST$_S$	98.19	97.58	95.43	94.09	98.53	97.12	93.91	95.35	93.79	94.65
+ST$_{S+T}$	–	97.30	94.97	94.08	–	–	93.83	95.33	93.64	94.72
WCON$_S$	**98.49**	97.87	**96.61**	**94.99**	**98.77**	**97.49**	**94.56**	**95.68**	94.34	95.12
WCON$_{S+T}$	98.47	**98.13**	96.10	94.95	**98.77**	97.45	94.47	95.61	**94.47**	**95.28**

Table 3.12. Cross-domain performance of the model with and without additional unlabeled text. "ST" indicates self-training. Methods denoted with "S" and "$S + T$" used unlabeled text in source domains and that in both source and target domains, respectively.

domains, obtained auto-segmented text by applying the same baseline segmenter described in §3.3, and then trained word embeddings from auto-segmented text with the Word2Vec toolkit. Finally, we trained from scratch the WCON model initialized with learned word embeddings. We used resources in Table 3.11 as unlabeled text for target domains[24] in addition to BCCWJ-NC text for source domains used in previous experiments. For comparison, we also evaluated the baseline model enhanced with the same source and target unlabeled text by self-training.

Table 3.12 shows results for baseline and proposed models with and without

[24] For unlabeled text of journal, patent, recipe, and blog domains, we used computer science paper abstracts published on IPSJ Digital Library (`https://www.ipsj.or.jp/e-library/digital_library.html`), NTCIR-8 PATMT Test Collection (`http://research.nii.ac.jp/ntcir/permission/ntcir-8/perm-en-PATMT.html`), the Cookpad dataset (`https://www.nii.ac.jp/dsc/idr/cookpad/`), and Yahoo! Chiebukuro data (3rd edition) (`https://www.nii.ac.jp/dsc/idr/yahoo/chiebkr3/Y_chiebukuro.html`), respectively.

additional unlabeled text. Although the baseline model using self-training from source unlabeled text improved performance over the pure-supervised baseline, its performance was inferior to that of WCON with the same additional resource. Besides that, we did not obtain further improvements on most domains by adding target domain text to the baseline. The enhanced WCON model achieved performance similar to that of the original WCON model on many domains and also achieved improvements by more than 0.1 points on journal, dining, and travel domains. However, its performance greatly (0.5 points) decreased on the patent domain, perhaps because of poor quality of word embeddings in this domain, whose occurring words differ greatly from those of the source domain (Table 3.6). Additionally, the baseline segmenter might generate noisy inputs to train word embeddings. Thus, simple addition of unlabeled text in a target domain did not necessarily contribute to further improvement. A possible solution is iterating self-training steps; we can use a trained WCON model for training a new WCON model by generating higher quality of word embeddings, despite the high cost of doing so. Another prospective method for constructing a more reliable vocabulary is to combine annotated resources such as lexicon and unlabeled text in a target domain. We leave this for the future, however.

3.4.6. Segmentation Examples

To examine segmentation results of actual sentences by the different methods, we picked up sentence segments (a)–(l) from the JDC's target domain test sets. In Figure 3.4, we show WCON's results, in addition to BASE and CON's results for reference. Regarding parts of sentences, in Figure 3.5, we also show weight values α_{ij} learned by WCON.

WCON predicted correct segmentation for (a)–(f) but predicted wrong results for (g)–(l). Also, as shown in Figure 3.5, the model attended to proper words for most characters in the former correct examples (attention accuracy for each segment ranged from 66–100%), but it often failed to pay correct attention in the latter incorrect examples (accuracy 0–50%). Besides that, we examined segmentation results by the oracle model WCON_O, in which attention weights are set to 1 for gold words (if existing in the vocabulary) and 0 for other word candidates, while other parameters are fixed to the original values of the trained

	(a)	(b)	(c)	(d)	(e)	(f)
BASE	も |しくは	多言 |語音 |声	干しし |い |た |け	色収 |差係 |数	未定 |着	しょう |が |汁
CON	もしくは	多 |言語 |音声	干し |しいたけ	色収 |差係 |数	未定 |着	しょうが汁
WCON	もしくは	多 |言語 |音声	干し |しいたけ	色 |収差 |係数	未 |定着	しょうが |汁
Gold	もしくは ⟨*moshikuwa*⟩ (or)	多 |言語 |音声 ⟨*ta*|*gengo*|*onsē*⟩ (multi|lingual|speech)	干し |しいたけ ⟨*hoshi*|*shītake*⟩ (dried|shiitake)	色 |収差 |係数 ⟨*iro*|*shūsa*|*kēsū*⟩ (color|abrration|coefficient)	未 |定着 ⟨*mi*|*tēchaku*⟩ (un-|fixed)	しょうが |汁 ⟨*shōga*|*jiru*⟩ (ginger|juice)

	(g)	(h)	(i)	(j)	(k)	(l)
BASE	表現 |物	半 |円筒 |状	セグメンテーション	ほうれん草	下 |茹 |で	めん |つ |ゆで
CON	表 |現物	半円 |筒状	セグメンテーション	ほうれん草	下 |茹 |で	めんつ |ゆで
WCON	表 |現物	半円 |筒状	セグ |メンテーション	ほう |れん草	下 |茹 |で	めんつ |ゆで
WCON_O	表現 |物	半 |円筒 |状	セグメンテーション	ほうれん |草	下 |茹 |で	めんつゆ |で
Gold	表現 |物 ⟨*hyōgen*|*butsu*⟩ (expression)	半 |円筒 |状 ⟨*han*|*entō*|*jō*⟩ (in the form of|half|cylinder)	セグメンテーション ⟨*segumentēshon*⟩ (segmentation)	ほうれん |草 ⟨*horēn*|*sō*⟩ (spinach)	下 |茹で ⟨*shita*|*yude*⟩ (preparatory|boiling)	めんつゆ |で ⟨*mentsuyu*|*de*⟩ (with|noudle soupe base)

Figure 3.4. Examples of segmentation results by models and gold segmentation in the BCCWJ test set. Correct results are highlighted with gray background color.

WCON model. The oracle model predicted correct segmentation for most sentences that the original model incorrectly segmented. These observations, along with performance improvement results by the "probabilistic oracle model" discussed in §3.4.4, suggest that the model generated segmentation results depending on calculated attention weights. Note that the vocabulary does not contain the gold words (セグメンテーション and 茹で) for (i)[25] and (k) among the examples of the oracle model's incorrect segmentation.

Segments (g), (h), and (i) were correctly segmented by BASE, but not by WCON. Such cases can be improved by emphasizing baseline segmentation, and it may be effective to use word information only when the maximum attention weights for candidate words are sufficiently high in WCON.

3.5. Related Work

The Use of Words in Character-Based Word Segmentation. Recent work on Chinese word segmentation has utilized word information on a character-based framework. Using word boundary information from auto-segmented text, for instance, Zhou et al. [173] pre-trained character embeddings. Wang and Xu

[25] The gold word セグメンテーション was in the original vocabulary without the length limit, but the trained model with the maximum word length of 4 excluded the word.

w_j ＼ x_i	干	···	干し ⟨*hoshi*⟩ (dried)	しい ⟨*shī*⟩	···	しいたけ ⟨*shītake*⟩ (shītake)
干	0.28		**0.72**	–		–
し	–		**0.99**	–		–
し	–		–	0.21		**0.71**
い	–		–	**0.92**		0.03
た	–		–	–		**0.99**
け	–		–	–		**1.00**

(c)

w_j ＼ x_i	色 ⟨*iro*⟩ (color)	収 ⟨*shū*⟩	···	数 ⟨*sū*⟩ (number)	収差 ⟨*shūsa*⟩ (aberra -tion)	係数 ⟨*kēsū*⟩ (coeffi -cient)
色	**1.00**	–		–	–	–
収	–	0.03		–	**0.97**	–
差	–	–		–	**0.99**	–
係	–	–		–	–	**1.00**
数	–	–		0.08	–	**0.92**

(d)

w_j ＼ x_i	···	セグ ⟨*segu*⟩	テー ⟨*men*⟩	ショ ⟨*sho*⟩	メンテ ⟨*mente*⟩ (mainte -nance)	ション ⟨*shon*⟩
セ		**0.63**	–	–	–	–
グ		**0.95**	–	–	–	–
メ		–	–	–	**0.99**	–
ン		–	–	–	**1.00**	–
テ		–	0.00	–	**0.93**	–
ー		–	**0.88**	–	–	–
シ		–	–	**0.61**	–	0.12
ョ		–	–	**0.49**	–	0.42
ン		–	–	–	–	**0.54**

(i)

w_j ＼ x_i	···	草 ⟨*sō*⟩ (grass)	ほう ⟨*hō*⟩ (toward)	ほうれ ⟨*hōre*⟩	れん草 ⟨*rensō*⟩	ほうれん ⟨*hōrensō*⟩ (spinach)
ほ		–	0.18	**0.71**	–	0.09
う		–	**0.83**	0.08	–	0.04
れ		–	–	0.08	**0.71**	0.13
ん		–	–	–	**0.91**	0.04
草		**0.69**	–	–	0.28	–

(j)

Figure 3.5. Weight α_{ij} learned by WCON for sentences (c), (d), (i), and (j) in Table 3.4. Weights to gold words are highlighted with gray background.

[142] explicitly introduced word information into their CNN-based model and concatenated embeddings of a character and multiple words corresponding to n-grams (n ranging from 1–4) that include the target character. Moreover, Yang et al. [157] proposed a lattice LSTM model with subsequence (i.e., word or subword) information. Their model integrates information on a character and the word ending with the character into an LSTM cell vector for the character using a gate-mechanism.

After publishing/submitting our original work, Tian et al. [132] proposed a memory network that incorporates word information into character-level segmentation, which was similar to our model but achieved better performance than our own. In contrast to our work, they constructed a word lexicon using unsupervised wordhood measures such as accessor variety [29], introduced an additional parameter matrix to multiply the attention-weighted sum of word vectors, used position embeddings to encode the relationship between a current character and a target word, and used pre-trained BERT or ZEN [26] character embeddings.

Semi-Supervised Learning for Word Segmentation. Especially to improve performance on OOV words, semi-supervised learning with unlabeled data has been explored for word segmentation. Typical approaches include self-training [66], co-training [163], statistical features [127] such as accessor variety, and frequent substrings [117]. As a common practice in recent neural models, large unlabeled text has been used to pre-train character, subword, or word embeddings [14, 157, 166, 172].

Attention Mechanism. An attention mechanism [6, 69] was first introduced in MT to focus on appropriate parts of a source sentence during decoding. This mechanism has been widely applied to various NLP tasks, including question answering [125], relation extraction [63], and natural language inference [98]. To determine the relative importance of a word itself and characters inside the word, Rei et al. [106] introduced a gate-like attention mechanism on their word-based sequence labeling model.

3.6. Conclusion

Aiming to contribute to disambiguating word boundaries, we proposed a word segmentation model that integrated word-level information into a character-based framework. Experimental results show that our model with attention-based composition functions achieved better performance than model variants without attention and competitive performance to existing Chinese and Japanese segmentation models. The main findings from our analysis are, first, word information from auto-segmented text alleviated the unknown word problem and also contributed to robust performance for cross-domain segmentation. Second, the attention mechanism learned appropriate weights for words, leading to accurate segmentation. Third, because of learned attention weights, our model can generate intuitively interpretable segmentation results.

4. Auxiliary Lexicon Word Prediction for Cross-Domain Word Segmentation

4.1. Introduction

As described in §3.1, neural network models have been widely applied to word segmentation in recent years. Those neural models demonstrated large performance improvements for in-domain word segmentation. However, those models are based on supervised learning and require a large amount of manually labeled data to obtain satisfactory performance. Therefore, a main challenge remains 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 can be exploited as complementary resources, which can be collected or constructed more easily than fully-labeled data. A lexicon feature [92, 146, 165] 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 [79], uses pseudo-labeled data generated from unlabeled data and a lexicon. Liu et al. [68] and Zhao et al. [171] 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 chapter, we propose a 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 work are as follows:

- We introduced a word segmentation method to learn explicit signals of word occurrences with surrounding contexts from unlabeled sentences.
- We demonstrated that the method improved performance for various target domains, while preventing performance degradation for source and other domains.
- Our model achieved better or competitive performance on Japanese and Chinese datasets, compared with existing methods for cross-domain word segmentation.

4.2. Proposed Method

In addition to labeled data $\mathcal{D}_l = \{(\boldsymbol{x}, \boldsymbol{t})\}$, we assume that unlabeled data $\mathcal{D}_u = \{\boldsymbol{x}\}$ and a word lexicon $\mathcal{L} = \{w\}$ are available. A word w in a lexicon is a sequence of characters. The j-th character in a word w is denoted as $w[j]$ and the length of a word as $|w|$. We introduce the proposed auxiliary task and task-specific multi-layer perceptrons (MLPs) into the baseline BiLSTM-softmax[26] model in §2.3, but these can be integrated into any neural architectures, including CNNs and SANs.

[26] We did not adopt the CRF-based prediction because it did not saliently outperform softmax-based prediction in our preliminary experiments.

Lexicon:

$\mathcal{L}$ = {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:

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

Auxiliary label sequences:

$\boldsymbol{u}^B$	=	(	1	0	0	1	0	0	0	1	0	1	0	)
$\boldsymbol{u}^I$	=	(	0	1	1	1	1	0	0	0	0	0	0	)
$\boldsymbol{u}^E$	=	(	0	0	1	0	0	1	0	0	1	0	1	)
$\boldsymbol{u}^S$	=	(	0	0	0	0	0	0	1	1	1	0	0	)

Figure 4.1. An example of auxiliary label sequences $\boldsymbol{u}^B$, $\boldsymbol{u}^I$, $\boldsymbol{u}^E$, and $\boldsymbol{u}^S$ of the `B`, `I`, `E`, and `S` positions for a lexicon $\mathcal{L}$ and a sentence $\boldsymbol{x}$, which means 'create a database.'

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.

The 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, namely, `B`, `I`, `E`, and `S` that were defined in §2.1.1. Formally, an auxiliary label sequence $\boldsymbol{u}^B = \boldsymbol{u}^B_{1:n} \in \{0,1\}^n$ is generated for each (labeled or unlabeled) sentence $\boldsymbol{x} = \boldsymbol{x}_{1:n}$ by matching substrings of $\boldsymbol{x}$ with any words in a lexicon $\mathcal{L}$. Given a character position j and a word length k such that $j = 1$ and $k > 1$, an auxiliary label u^B_i for a character x_i indicates whether x_i corresponds to the beginning of a word w, and it is defined as follows:

$$u^B_i = \begin{cases} 1 & (\exists w \in \mathcal{L}, |w| = k \text{ and } x_i = w[j]) \ , \\ 0 & (\text{otherwise}) \ . \end{cases} \tag{4.1}$$

Similarly, auxiliary label sequences $\boldsymbol{u}^I$, $\boldsymbol{u}^E$, and $\boldsymbol{u}^S \in \{0,1\}^n$ are generated to indicate the inside of a word, the end of a word, and a single character word. Each label $\boldsymbol{u}^I_i$, $\boldsymbol{u}^E_i$, and $\boldsymbol{u}^S_i$ is similarly defined by letting $1 < j < k$, $j = k > 1$, and $j = k = 1$ in Eq. (4.1), respectively. Figure 4.1 illustrates an example of auxiliary label sequences for a sentence and a lexicon. For example, $(u_3^B, u_3^I, u_3^E, u_3^S) = (0,1,1,0)$ because $x_3 =$タ is the inside character of データベース and the end character of データ.

The loss of the segmentation task is defined as $L_{\text{seg}} = L_{\text{softmax}}$ in Eq. (2.2). Similarly, the loss of the auxiliary task is defined for each position by the cross entropy based on auxiliary labels. Then, the auxiliary loss L_{aux} is the sum of the losses for four LWP tasks with respect to `B`, `I`, `E` and `S` positions:

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

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

$$L_{\text{seg}}(\mathcal{D}_l) + \lambda L_{\text{aux}}(\mathcal{D}_l \cup \mathcal{D}_u) \ , \tag{4.2}$$

where λ is a hyperparameter to control the importance of the auxiliary task. 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 MLPs to learn task-specific representations. Let d_m be a hyperparameter. A hidden vector $\boldsymbol{h}_i$ from the BiLSTM layers is transformed into task-specific vectors $\boldsymbol{m}_i, \boldsymbol{l}_i \in \mathbb{R}^{d_m}$ via different MLPs with one hidden layer for the main task (MLP$_{\text{seg}}$) and the auxiliary task (MLP$_{\text{aux}}$):

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

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

$$\begin{aligned} \boldsymbol{s}_i &= W_s \boldsymbol{m}_i + \boldsymbol{b}_s \ , \\ \boldsymbol{s}_i^p &= W_{a,p} \boldsymbol{l}_i + \boldsymbol{b}_{a,p} \ , \end{aligned}$$

where $W_s \in \mathbb{R}^{|\mathcal{T}| \times d_m}$, $W_{a,p} \in \mathbb{R}^{2 \times d_m}$, $\boldsymbol{b}_s \in \mathbb{R}^{|\mathcal{T}|}$, and $\boldsymbol{b}_{a,p} \in \mathbb{R}^2$ are trainable parameters. The model outputs predicted label distributions for the segmentation and auxiliary tasks similarly to the baseline method described in §2.3, and the loss is calculated by Eq. (4.2).

4.3. Experimental Settings

4.3.1. Language Resources

Datasets. For Japanese experiments, we used the Japanese Dependency Corpus (JDC), which consists of six domain datasets from different data sources. We used journal (JNL), patent (JPT), and recipe (RCP) domain data as out-of-domain test sets and used sentences in the remaining three domains as source domain data, which we called GEN. We selected the same 500 development and 2,000 (in-domain) test sentences from the source domain data, as those used in the experiments in Chapter 3.

For Chinese experiments, we used two source domain data: Chinese Treebank 5.0 (CTB5)[27] and the SIGHAN Bakeoff 2005 [28] PKU data. As evaluation data for CTB5, we used an internet novel dataset ZhuXian (C-ZX) [165]. As evaluation data for PKU, we used three internet novel datasets [103], ZhuXian (P-ZX), FanRenXiuXianZhuan (FR), and DouLuoDaLu (DL) together with two science and technology datasets [102] 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 [103, 165]. We followed the training/development/test split of CTB5 by Zhang and Clark [169] 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 4.1 shows the dataset statistics. Values in the train, dev, test, and unlabeled rows indicate the numbers of sentences, and values in the lexicon row

[27] https://catalog.ldc.upenn.edu/LDC2005T01

	Japanese (JDC)				Chinese (CTB5)		Chinese (PKU)					
	Source	Target			Source	Target	Source	Target				
	GEN	JNL	JPT	RCP	CTB5	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
UL	480K	480K	480K	480K	480K	27K	480K	27K	130K	40K	46K	
Lex	570K	13K	134K	21K	390K	0.5K	390K	0.5K	0.7K	0.5K	0	0

Table 4.1. Dataset statistics. “UL” indicates unlabeled data. “Lex” indicates lexicon.

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 non-core data of BCCWJ version 1.1 for GEN and Chinese Gigaword Fifth Edition for CTB5 and PKU.

As unlabeled data for cross-domain experiments, we used Japanese computer science paper abstracts published on IPSJ Digital Library for JNL, NTCIR-8 PATMT Test Collection for JPT, the “steps” portion of the Cookpad Dataset for RCP. We used the unlabeled data provided by Ye et al.[28] [160] 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 UniDic (unidic-mecab-2.1.2[29]) and Jieba dictionary[30] 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 thesaurus[31] 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

[28] https://github.com/vatile/CWS-NAACL2019

[29] https://ccd.ninjal.ac.jp/unidic/back_number

[30] https://github.com/fxsjy/jieba/blob/master/jieba/dict.txt

[31] https://dbarchive.biosciencedbc.jp/en/mecab/data-1.html

name lexicon[32] 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.[33] 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 conjunctions[34] 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.

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.

4.3.2. Baseline Methods

We used the following three baselines.

- A naïve baseline (BASE): A BiLSTM model, which is described in §2.3, trained from source labeled data.

- A self-training baseline (ST): A BiLSTM model trained from labeled data in

[32] `https://github.com/egrcc/Cross-Domain-CWS/blob/master/dataset/preprocess_data/zx/zx_dict.txt`

[33] 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. For example, 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."

[34] For example, an original description "牛肉または豚肉" (beef or pork) is split into "牛肉" and "豚肉."

source domain and auto-segmented data in each target domain. To obtain auto-segmented data, BASE was applied to target unlabeled 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-sized[35] vector $\boldsymbol{l}_i$ was constructed for a character x_i and used $\boldsymbol{e}'_i = \boldsymbol{e}_i \oplus \boldsymbol{l}_i$ as input to BiLSTM layers, instead of the character embedding $\boldsymbol{e}_i$. Differing from Zhang et al.'s work [167] that extended a standard LSTM architecture to incorporate lexicon features, these features were used in the above simple manner.

4.3.3. 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. (4.2) 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 4.2 gives the hyperparameters for the baseline and proposed methods. We used lexical words whose length was less or equal to six when generating auxiliary labels, because there were many cases in which longer lexical entries in target lexicons were not single words.[36] We applied dropout [161] 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

[35] The dimension was $15 = 5$ positions × 3 length groups.

[36] 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).

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 4.2. Hyperparameter values for the baseline and proposed methods.

epoch after the first five epochs. We trained models for up to 20 epochs and used early stopping based on the F_1 score on the development set.

4.4. Results and Analysis

4.4.1. 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 4.3 shows the mean F_1 score of three runs for each method and each dataset. We conducted McNemar's tests on the differences between word-level predictions (TP or FN) of two systems for gold words. The symbols $\star$, †, and ‡ in Table 4.3 indicate statistical significance at the 0.001 level over BASE, ST, and LF, respectively. The symbol $\underline{‡}$ indicates that the performance is significantly lower than that of LF.

The improvements of ST and LWP-S over BASE were significant on JDC and PKU, and those of LF over BASE were significant on three datasets. The performance differences between ST and LWP-S were significant on three domains,

Method	Resource	JDC	CTB5	PKU
BASE	–	98.0	96.9	94.5
ST	$\mathcal{U}_s$	$98.1^{\star}$	96.8	$94.6^{\star}$
LF	$\mathcal{L}_s$	$\mathbf{98.5}^{\star}$	$\mathbf{97.0}^{\star}$	$\mathbf{95.6}^{\star}$
LWP-S	$\mathcal{U}_s$, $\mathcal{L}_s$	$98.4^{\star\dagger}$	$96.7^{\dagger}$	$95.3^{\star\dagger\ddagger}$

Table 4.3. Performance on the source domain test sets. The resource column lists resources used by each method: source unlabeled data $\mathcal{U}_s$ and source lexicon $\mathcal{L}_s$.

Position	JDC	CTB5	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)

Table 4.4. Accuracy of LWP-S on auxiliary label classification on the test sets. Values in "()" indicate the percentage of positive labels ($u_i^p = 1$) in all labels for each position $p \in \{\text{B, I, E, S}\}$.

and that between LF and LWP-S was significant only on PKU.[37] Compared to LF, the proposed method achieved similar but slightly lower segmentation performance; 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 4.4 shows the mean accuracy of three runs of 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.

[37] Our recall-oriented significance tests showed that the improvement of LWP-S (recall of 97.6) over ST (97.3) on CTB5 was significant, although ST performed better in terms of F_1 score as in Table 4.3.

Method	Resource	JNL	JPT	RCP	C-ZX	P-ZX	FR	DL	DM	CPT
BASE	–	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$^{\underline{\star}}$	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

Table 4.5. Performance on the target domain test sets. 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).

4.4.2. 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 4.5 shows the mean F_1 score of three runs for each method and each dataset. The symbols $\star$, $\dagger$, and $\ddagger$ in Table 4.5 indicate statistical significance at the 0.001 level over BASE, ST, and LF, respectively, according to the McNeamer's tests similar to the in-domain experiments. The symbol $\underline{\star}$ indicate that the performance is significantly lower than that of BASE.

ST showed limited improvements (+0.2 points over BASE on average). LF showed a more clearly improved performance (+2.0 points over BASE). The proposed method, LWP-T, achieved larger improvements (+3.2 points over BASE) than the other enhanced baselines on all domains. These results validated that our auxiliary task enabled to learn word indicators in target contexts. The performance of ST, LF, and LWP-T was significantly better than that of BASE on six, eight, and nine out of nine datasets, respectively.[38] Moreover, the performance of LWP-T was significantly better than that of ST and LF on all domains. Note that the performance of LWP-S was also significantly better than that of BASE and ST on all domains and than that of LF on the three Japanese domains.

Table 4.6 shows the OOV rate of each test set when given a vocabulary $\mathcal{V}$.

[38] Our significance tests showed that the improvement of ST (recall of 94.9) over BASE (94.8) on JPT, and the degradation of LF (recall of 94.5) over BASE (95.0) on RCP were significant, although BASE performed better in the former case and LF performed better in the latter case in terms of F_1 score as in Table 4.5.

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

Table 4.6. OOV rates of the test sets for a vocabulary. $\mathcal{V}_{w:\text{train}}$ indicate the set of gold words in the corresponding training set. Δ indicates the difference of the rates between $\mathcal{V}_t$ and $\mathcal{V}_0$.

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.[39]

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 more than 3 point improvements in F_1 and more than 11 point reduction in OOV rate on C-ZX, P-ZX, and DM, more than 2.5 point improvements in F_1 and more than 7 point reduction in OOV rate on JPT, FR, DL, and CPT, and about 1 point improvements in F_1 and about 5 or 6 point reduction in OOV rate on JNL and RCP. Second, the proposed method is not sensitive to the size of lexicons and unlabeled data; we observed large improvements on the Chinese domains in spite of the small size of lexicons and unlabeled data as shown in Table 4.1, probably owing to the OOV rate reduction as above. This also suggests that a reasonable size of lexicons and 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 4.5. The higher performance of LWP-O demonstrates that the proposed method can achieve further

[39] For example, the OOV rate of a test set consisting of six word tokens, A, A, B, C, D, and E for a vocabulary $\mathcal{V} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ is $2/6 = 33.3\%$.

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+ '11 [92] ○		✓	97.8	97.2	**95.5**	–	–	–	–	–	–
Kitagawa+ '18 [53]			97.6	93.1	94.0	–	–	–	–	–	–
Higashiyama+ '19 [39]	✓		98.1	96.7	95.2	–	–	–	–	–	–
Liu+ '14 [68] ○	✓	✓	–	–	–	90.6	–	–	–	–	–
Zhou+ '17 [173]	✓		–	–	–	90.1	–	–	–	–	–
Zhao+ '18 [171]	✓	✓	–	–	–	**92.9**	–	–	–	–	–
Zhang+ '10 [168] ○			–	–	–	–	86.8	85.9	90.5	77.9	84.6
Ye+ '19 [160]	✓		–	–	–	–	89.6	89.6	**93.5**	82.2	85.1
Gan+ '19 [31]	✓		–	–	–	–	**90.5**	**91.1**	93.0	–	–

Table 4.7. Comparison with state-of-the-art methods on the test sets. "UL" and "lex" indicates whether a method uses additional unlabeled data and lexicons, respectively. Non-neural methods are marked with the symbol ○.

improvements using a higher-coverage lexicon.

4.4.3. Comparison with State-of-the-Art Methods

Table 4.7 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 [168] is cited from Ye et al. [160]). We cited the results of Gan and Zhang's [31] 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, including the proposed method in Chapter 3 [39], 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.'s [171] method that incorporated partially-labeled target sentences (F_1 of 91.6 on C-ZX) but their method obtained further gains (+1.3 points as in Table 4.7) by integrating a character-level LM. Similarly, Gan and Zhang [31] showed improvements by introducing BERT character embeddings. Our method may also obtain benefits from combining language modeling-like information learned from a huge amount of data.

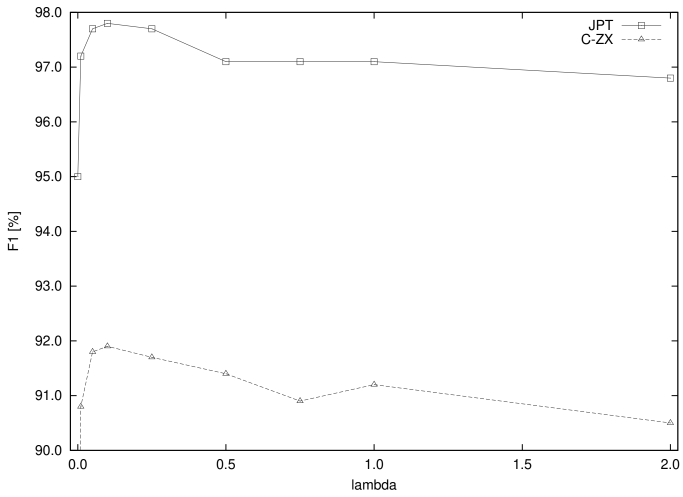


Figure 4.2. Performance of LWP-T for each λ value on the JPT and C-ZX test sets.

4.4.4. Influence of Weight for Auxiliary Loss

We investigated the influence of the hyperparameter λ to control the importance of LWP task in Eq. (4.2). Figure 4.2 shows F_1 scores of single runs of 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.

4.4.5. 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 4.8 shows the mean F_1 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 for CTB5) compared with the models adapted to other domains; (2) the models adapted to

Method	A\E	GEN	JNL	JPT	RCP
BASE	GEN	98.0	97.2	95.0	94.3
LWP-S	GEN	**98.4**	97.8	97.0	95.3
LWP-T	JNL	98.1	**98.2**	97.4	94.8
	JPT	98.2	97.7	**97.6**	94.8
	RCP	98.2	97.2	95.8	**95.4**

(i) JDC

Method	A\E	CTB5	C-ZX
BASE	CTB5	**96.9**	86.3
LWP-S	CTB5	96.7	88.4
LWP-T	C-ZX	96.5	**91.7**

(ii) CTB5

Method	A\E	PKU	P-ZX	FR	DL	DM	CPT
BASE	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**

(iii) PKU

Table 4.8. Performance of models adapted to adaptation domains (A) on the test sets of evaluation domains (E). Cells with gray background indicate the results on the same adaptation and evaluation domains.

any target domains performed better than BASE on the source domains (except for CTB5) and performed similarly to or better than BASE 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.

4.4.6. Performance for Out-of-Training-Vocabulary Words

Overall Results. We examined the performance of the proposed method on out-of-training-vocabulary (OOTV) words, that is, words that are not in the training set. Table 4.9 shows the mean recall of three runs of each method for all OOTV words in each test set. The proposed method, LWP-T, performed the best on eight out of nine domains and the improvements over BASE and LF were +7.1 and +5.7 points on average. The performance of LWP-S was in between that of BASE and LWP-T on those eight domains. The performance difference

Method	JNL	JPT	RCP	C-ZX	P-ZX	FR	DL	DM	CPT
BASE	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**

Table 4.9. Recall of all OOTV words in each test set.

Group	In $\mathcal{L}$	In $\mathcal{U}$	JNL	JPT	RCP	C-ZX	P-ZX	FR	DL	DM	CPT
II	✓	✓	81.2	68.0	82.3	76.6	71.7	65.3	70.7	45.5	43.6
IO	✓		0.3	1.1	1.5	0.1	0.0	0.0	0.0	0.0	0.0
OI		✓	14.0	26.6	11.9	23.4	28.2	34.6	29.2	54.4	56.4
OO			4.5	4.4	4.4	0.0	0.1	0.1	0.1	0.1	0.0

(i) Percentage of each word group in all OOTV words.

Group	In $\mathcal{L}$	In $\mathcal{U}$	JNL	JPT	RCP	C-ZX	P-ZX	FR	DL	DM	CPT
II	✓	✓	93.2	93.8	85.3	89.7	85.8	89.8	88.9	92.5	90.3
IO	✓		33.3	79.8	79.2	100.0	–	–	–	–	–
OI		✓	80.1	80.0	77.6	28.2	50.4	12.8	4.7	19.1	42.0
OO			74.4	72.6	58.7	–	0.0	0.0	0.0	0.0	–

(ii) Recall of LWP-T for each word group.

Table 4.10. Results for OOTV word groups based on data inclusion. Cells corresponding to groups smaller than 1% are shown with gray background.

between LWP-T and LF suggests the importance of learning word information within the target domain's context for accurate recognition of OOTV words.

Results for OOTV Word Groups. As shown in Table 4.10, we then evaluated the mean recall of three runs of LWP-T for OOTV words divided into four groups {II, IO, OI, OO} according to whether the words were (I) or were not (O) contained in the lexicon $\mathcal{L} = \mathcal{L}_s \cup \mathcal{L}_t$ and whether words did (I) or did not (O) occur in the unlabeled data $\mathcal{U} = \mathcal{U}_s \cup \mathcal{U}_t$. More than 94% of OOTV words were in group II or OI, that is, most OOTV words occurred at least once in the unlabeled data. The proposed method achieved much higher recall of at least 85% for words in group II than those in group OI, particularly in the Chinese domains; in addition, the method partially recognized words not in the lexicon,

	Group	JNL	JPT	RCP	C-ZX	P-ZX	FR	DL	DM	CPT
%	R=100	87.2	82.7	77.0	69.5	61.2	61.3	61.8	46.2	58.7
	0<R<100	2.7	6.1	6.2	5.9	4.4	6.3	4.1	3.4	6.7
	R=0	10.1	11.2	16.8	24.6	34.4	32.3	34.1	50.4	34.6
Average length	R=100	4.1	3.0	3.1	2.1	2.3	2.3	2.3	2.5	2.8
	0<R<100	4.0	2.5	2.8	2.1	2.3	2.1	2.3	2.9	2.5
	R=0	5.9	3.8	3.3	2.6	2.4	2.5	2.5	4.3	3.6
Non-kanji ratio	R=100	70.1	60.4	77.3	0.0	0.1	2.0	0.0	4.0	25.4
	0<R<100	76.0	60.0	75.3	0.7	0.8	0.7	0.0	14.1	31.1
	R=0	94.6	80.3	89.3	0.1	0.9	5.0	1.0	16.3	52.5

Table 4.11. Results for OOTV word groups based on recall (R) ranges.

although the recall greatly differed by language and domain, i.e, by 5%–80%. From a comparison between the recall for groups II and IO, the method achieved the worse recall for group IO (except for C-ZX), which were in the lexicon but were not in the unlabeled data. As expected, the results show that occurrences in both lexicons and unlabeled data are necessary for segmenting OOTV words accurately when using the proposed method.

To analyze the words that were difficult to segment by LWP-T, we additionally investigated OOTV word types by grouping in terms of recall (R): R = 0, 0 < R < 100, and R = 100. For example, an OOTV word type 位相 *isō* 'phase' in the JPT test set is classified into the word group $0 < R < 100$ because it was correctly segmented 36 times out of 38 occurrences. Table 4.11 shows the percentages of word type groups in all OOTV words, as well as the average word length and non-kanji ratio of word types in each group. The non-kanji ratio indicates the percentage of word types containing a non-kanji character, such as hiragana, katakana, and Roman letters, in all word types in a group.[40] The recognition of more than 93% of OOTV words by the proposed method was either completely successful (R = 100) or a complete failure (R = 0). Words with R = 0 tend to have longer lengths and higher non-kanji ratios than those with R = 100, suggesting the difficulty of recognizing such words.

[40] Percentage, average word length, and non-kanji ratio were calculated based on the number of word types (not the total occurrences of word tokens).

Word		In $\mathcal{L}$	Freq	Recall		
				BASE	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**

Table 4.12. Recall of top-10 frequent OOTV word types in the JPT test set.

Word		In $\mathcal{L}$	Freq	Recall		
				BASE	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			1094	59.4	84.8	**95.8**

Table 4.13. Recall of top-10 frequent OOTV word types in the C-ZX test set.

Results for Frequent OOTV Words. We finally evaluated the performance on high frequency OOTV word types in the JPT, C-ZX, and FR test sets. Table 4.12–4.14 shows the mean recall of three runs of BASE, LF, or LWP-T on each OOTV word type, along with word frequencies and whether each word type was in the lexicon $\mathcal{L} = \mathcal{L}_s \cup \mathcal{L}_t$. The proposed method recognized OOTV 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 OOTV 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 five of the top-10 OOTV words in FR. This result suggests that the lexicon-based models are

Word		In $\mathcal{L}$	Freq	Recall		
				BASE	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

Table 4.14. Recall of top-10 frequent OOTV word types in the FR test set.

biased such that a character sequence without positive signals is not recognized as a word. These cases correctly segmented by BASE can be alleviated by combining both methods; for example, in the proposed method, it may be effective to add words segmented with high reliability by BASE to the lexicons.

4.5. Related Work

It has been demonstrated that the use of linguistic resources, such as unlabeled data, partially-labeled data, and lexicons, has achieved robust performance for out-of-domain texts. Well-known techniques using unlabeled data include self-training [66] and statistical features [123, 146] 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 [48, 68, 163]. Distantly-supervised data generated from unlabeled data and lexicons was also used as partially-labeled data [68, 171]. Another well-known technique based on lexicons is a lexicon feature that indicates occurrence of lexicon entries [92, 146, 165].

Much work [14, 166, 172] mainly focused on improvements of in-domain performance by using general large unlabeled text, such as news articles or Wikipedia, to pre-train character/word embeddings. On the other hand, Zhou et al. [173] and Ye et al. [160] 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 et al. [144] integrated auto-segmented label information by an unsupervised segmenter into a neural segmentation model to boost the 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. [167] 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. [171] 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 LM learned from unlabeled data and trained the model with a modified loss function to handle partially-labeled data. Liu et al. [65] 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 [31] 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.

After submitting our original work, Ding et al. [27] proposed a CNN-based model with shared and domain-specific encoders for cross-domain word segmentation, which achieved better performance than our method. They adversarially trained their model with labeled data in a source domain and distantly annotated data in a target domain, which was built using domain-specific word candidates obtained based on statistical measures, such as mutual information, entropy, and TF-IDF.

4.6. Conclusion

In this chapter, we proposed a cross-domain word segmentation method using unlabeled data and lexicons. 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 documents, recipes, and novels. The experimental results demonstrated that our auxiliary task improved performance for target domains by 3.2 F_1 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.

5. User-Generated Text Corpus for Evaluating Japanese Morphological Analysis and Lexical Normalization

5.1. Introduction

MA methods for well-formed text [57, 92] have been actively developed taking advantage of the existing annotated corpora of news domains, but they perform poorly on UGT. Additionally, because of the frequent occurrence of informal words, lexical normalization to identify standard word forms is another important task in UGT. Some work have been devoted to both tasks in Japanese UGT [51, 107, 108, 112] to achieve the robust performance for noisy text. Previous researchers have evaluated their own systems using in-house data created by individual researchers, and thus it is difficult to compare the performance of different systems and discuss what issues remain in these two tasks. Therefore, publicly available data is necessary for a fair evaluation of MA and normalization performance on Japanese UGT.

In this chapter, we present the blog and Q&A forum normalization corpus (BQNC),[41] which is a public Japanese UGT corpus annotated with morphological, normalization, and word category information. We have constructed the corpus under the following policies: (1) available and restorable; (2) compatible with the segmentation standard and POS tags used in the existing representative corpora; and (3) enabling a detailed evaluation of UGT-specific problems.

[41] Our corpus is available at `https://github.com/shigashiyama/jlexnorm`.

For the first requirement, we extracted and used the raw sentences in the blog and Q&A forum registers compiled by the non-core data of BCCWJ, in which the original sentences are preserved.[42] For the second requirement, we followed the SUW criterion of NINJAL, which is used in various NINJAL's corpora, including manually annotated sentences in BCCWJ. For the third requirement, we organized linguistic phenomena frequently observed in the two registers as word categories, and annotated each word with a category. We expect that this will contribute to future research to develop systems that manage UGT-specific problems. BQNC comprises sentence IDs and annotation information, including word boundaries, POS, lemmas, standard forms of non-standard word tokens, and word categories.

Using BQNC, we evaluated two existing methods: a popular Japanese MA toolkit called MeCab [57] and a joint MA and normalization method by Sasano et al. [112]. Our experiments and error analysis showed that these systems did not achieve satisfactory performance for non-general words. This indicates that our corpus would be a challenging benchmark for further research on UGT.

5.2. Overview of Word Categories

Based on our observations and the existing work [46, 52], we organized word tokens that may often cause segmentation errors into two major types with several categories as shown in Table 5.1. We classified each word token from two perspectives: the type of vocabulary to which it belongs and the type of variant form to which it corresponds. For example, ニホン *nihon* 'Japan' written in katakana corresponds to a *proper name* and a *character type variant* of its standard form 日本 written in kanji.

Specifically, we classified vocabulary types into *neologisms/slang*, *proper names*, *onomatopoeia*,[43] *interjections*, *(Japanese) dialect words*, *foreign words*, and *emoti-*

[42] Twitter could be a candidate for a data source. However, redistributing original tweets collected via the Twitter Streaming APIs is not permitted by Twitter, Inc., and an alternative approach to distributing tweet URLs has the disadvantage that the original tweets can be removed in the future.

[43] "Onomatopoeia" typically refers to both the phonomime and phenomime in Japanese linguistics literature, similar to ideophones. We follow this convention in this thesis.

Category	Example	Reading	Translation	Standard forms
Type of vocabulary:				
(General words)				
Neologisms/Slang	コピペ	*copipe*	copy and paste	
Proper names	ドラクエ	*dorakue*	Dragon Quest	
Onomatopoeia	キラキラ	*kirakira*	glitter	
Interjections	おお	*ō*	oops	
Dialect words	ほんま	*homma*	truly	
Foreign words	ＥＡＳＹ		easy	
Emoticons/AA	（＾－＾）			
Type of variant form:				
(Standard forms)				
Character type variants	カワイイ	*kawaī*	cute	かわいい,可愛い
Alternative representations	大きぃ	*ōkī*	big	大きい
Sound change variants	おいしーい	*oishīi*	tasty	おいしい,美味しい
Typographical errors	つたい	*tsutai*	tough	つらい,辛い

Table 5.1. Word categories in BQNC.

cons/ASCII art (AA), in addition to general words.[44] A common characteristic of these vocabularies, except for general words, is that a new word can be indefinitely invented or imported. We annotated word tokens with vocabulary type information, except for general words.

From another perspective, any word can have multiple variant forms. Because the Japanese writing system comprises multiple script types including *kanji* and two types of *kana*, that is, *hiragana* and *katakana*,[45] words have orthographic variants written in different scripts. Among them, non-standard *character type variants* that rarely occur in well-formed text but occur in UGT can be problematic, for example, a non-standard form カワイイ for a standard form かわいい *kawaī* 'cute'. Additionally, ill-spelled words are frequently produced in UGT. We further divided them into two categories. The first is *sound change variants* that have a phonetic difference from the original form and are typically derived by deletions, insertions, or substitutions of vowels, long sound symbols (*chōon* "ー"),

[44] We observed a few examples of other vocabulary types, such as Japanese archaic words and special sentence-final particles in our corpus, but we treated them as general words.

[45] Morphographic *kanji* and syllabographic *hiragana* are primarily used for Japanese native words (*wago*) and Japanese words of Chinese origin (Sino-Japanese words or *kango*), whereas syllabographic *katakana* is primarily used, for example, for loanwords, onomatopoeia, and scientific names. Additionally, Arabic numerals, Roman letters (*rōmaji*), and other auxiliary symbols are used in Japanese sentences.

long consonants (*sokuon* “っ”), and mora nasal (*hatsuon* “ん”), for example, おいしーい *oishīi* for おいしい *oishī* ‘tasty’. The second category is *alternative representations* that do not have a phonetic difference and are typically achieved by substitution among uppercase or lowercase kana characters, or among vowel characters and long sound symbols, for example, 大きぃ for 大きい *ōkī* ‘big’. Moreover, *typographical errors* can be seen as another type of variant form. We targeted these four types of non-standard forms for normalization to standard forms.

5.3. Corpus Construction Process

BQNC was constructed using the following steps. The annotation process was performed by the author.

(1) Sentence Selection We manually selected sentences to include in our corpus from the blog and Q&A forum registers in the BCCWJ non-core data. We preferentially extracted sentences that contained candidates of UGT-specific words, that is, word tokens that may belong to non-general vocabularies or correspond to non-standard forms. As a result, we collected more than 900 sentences.

(2) First Annotation Sentences in the non-core data have been automatically annotated with word boundaries and word attributes, such as POS and lemma. Following the BCCWJ annotation guidelines [93, 94] and UniDic [24], which is an electronic dictionary database designed for the construction of NINJAL’s corpora, we refined the original annotations of the selected sentences by manually checking them. The refined attributes were token, POS, conjugation type, conjugation form, pronunciation, lemma, and lemma ID. Additionally, we annotated each token with a word category shown in Table 5.1 and a standard form ID if the token corresponded to a non-standard form.

Table 5.2 shows two examples of annotated sentences. We annotated each non-standard token with a standard form ID denoted as “[lemma ID]:[lemma](_[pronunciation])”, which is associated with the set of acceptable standard forms shown in Table 5.3.

Token	Translation	Standard form ID
イイ	good	38988:良い
歌	song	
です	(polite copula)	
ねェ	(emphasis marker)	28754:ね
ヨカッ	good	38988:良い_ヨカッ
タ	(past tense marker)	21642:た

Table 5.2. Examples of annotated text "イイ歌ですねェ" 'It's a good song, isn't it?' and "ヨカッタ" 'It was good.' Attributes except for token and standard form ID are abbreviated.

Standard form ID	Standard forms
21642:た	た
28754:ね	ね
38988:良い	良い,よい,いい
38988:良い_ヨカッ	良かっ,よかっ

Table 5.3. Examples of standard form IDs.

(3) Second Annotation We rechecked all tokens in the sentences that we finished the first annotation and fixed the annotation criteria, that is, the definitions of vocabulary types and variant form types, and standard forms for each word. Through these steps, we obtained 929 annotated sentences.

5.4. Detailed Definition of Word Categories

5.4.1. Type of Vocabulary

Through the annotation process, we defined the criteria for vocabulary types as follows.

Neologisms/Slang: a newly invented or imported word that has come to be used collectively. Specifically, we used a corpus reference application called Chunagon[46] and regarded a word as a *neologism/slang* if its frequency in BCCWJ

[46] https://chunagon.ninjal.ac.jp

was less than five before the year 2000 and increased to more than ten in 2000 or later.[47]

Proper Names: following the BCCWJ guidelines, we regarded a single word that corresponded to a proper name, such as person name, organization name, location name, and product name, as a *proper name*. In contrast to the BCCWJ guidelines, we also regarded an abbreviation of a proper name as a *proper name*, for example, ドラクエ in Table 5.1.

Onomatopoeia: a word corresponds to onomatopoeia. We referred to a Japanese onomatopoeia dictionary [154] to assess whether a word is onomatopoeic. We followed the criteria in the BCCWJ guidelines on what forms of words are onomatopoeic and what words are associated with the same or different lemmas.

Interjections: a word whose POS corresponds to an interjection. Although we defined standard forms for idiomatic greeting expressions registered as single words in UniDic,[48] we did not define standard and non-standard forms for other interjections that express feelings or reactions, for example, ええ *ē* 'uh-huh' and うわあ *uwā* 'wow'.

Foreign Words: a word from non-Japanese languages. We regarded a word written in scripts in the original language as a *foreign word*, for example, English words written in the Roman alphabet such as "plastic." Conversely, we regarded loanwords written in Japanese scripts (hiragana, katakana, or kanji) as general words, for example, プラスチック 'plastic.' Moreover, we did not regard English acronyms and abbreviations written in uppercase letters as foreign words because such words are typically also written in the Roman alphabet in Japanese sentences, for example, ＳＮＳ 'SNS.'

Dialect Words: a word from a Japanese dialect. We referred to a Japanese dialect dictionary [113] and regarded a word as a *dialect word* if it corresponded to

[47] The original sentences were from posts published between 2004 and 2009.

[48] Eight greeting words exist, for example, ありがとう *arigatō* 'thank you' and さようなら *sayōnara* 'see you.'

an entry or occurred in an example sentence. We did not consider normalization from a dialect word to a corresponding word in the standard Japanese dialect.

Emoticons/AA: nonverbal expressions that comprise characters to express feelings or attitudes. Because the BCCWJ guidelines does not explicitly describe criteria on how to segment emoticon/AA expressions as words, we defined criteria to follow emoticon/AA entries in UniDic.[49]

5.4.2. Type of Variant Form

There are no trivial criteria to determine which variant forms of a word are standard forms because most Japanese words can be written in multiple ways. Therefore, we defined standard forms of a word as all forms whose occurrence rates were approximately equal to 10% or more in BCCWJ among forms that were associated with the same lemma. For example, among variant forms of the lemma 面白い *omoshiroi* ‘interesting’ or ‘funny’ that occurred 7.9K times, major forms 面白い and おもしろい accounted for 72% and 27%, respectively, and other forms, such as オモシロイ and オモシロい, were very rare. In this case, the standard forms of this word are the two former variants. We annotated tokens corresponding to the two latter non-standard forms with the standard form IDs and the types of variant forms. We defined criteria for types of variant forms as follows.

Character Type Variants: among the variants written in different scripts, we regarded variants whose occurrence rates were approximately equal to 5% or less in BCCWJ as non-standard forms of *character type variants*. Specifically, variants written in kanji, hiragana, or katakana for native words and Sino-Japanese words, variants written in katakana or hiragana for loanwords, variants written in uppercase or lowercase Roman letters for English abbreviations are candidates for character type variants. We assessed whether these candidates were non-standard forms based on the occurrence rates.

[49] For example, if characters expressing body parts were outside of punctuation expressing the outline of a face, the face and body parts were segmented, but both were annotated with *emoticons/AA*, for example, “m（．＿．）m” → “m|（．＿．）|m.”

Alternative Representations: a form whose internal characters are (partially) replaced by special characters without phonetic differences. Specifically, non-standard forms of *alternative representations* include native words and Sino-Japanese words written in historical kana orthography (e.g., 思ふ for 思う *omō*/*omou* 'think'), and loanwords written as an unusual[50] katakana sequence (e.g., オオケストラ for オーケストラ 'orchestra'). Additionally, *alternative representations* include substitution with respect to kana: substitution of the long vowel kana by the long sound symbol (e.g., おいし〜 for おいしい *oishī* 'tasty'), substitution of upper/lowercase kana by the other case (e.g., ゎたし for わたし *watashi* 'me'), and phonetic or visual substitution of kana characters by Roman letters and symbols (e.g., かわＥ for かわいい *kawaī* 'cute' and こωにちは for こんにちは *konnichiwa* 'hello').

Sound Change Variants: a form whose pronunciation is changed from the original form. Specifically, *sound change variants* include the insertion of special moras (e.g., 強ーい *tsuyōi* for 強い *tsuyoi* 'strong'), deletion of moras (e.g., くさ *kusa* for くさい *kusai* 'stinking'), and substitution of characters/moras (e.g., っす *ssu* for です *desu* polite copula and すげえ *sugē* for すごい *sugoi* 'awesome').

Typographical Errors: a form with typographical errors derived from character input errors, kana-kanji conversion errors, or the user's incorrect understanding. For example, つたい *tsutai* for つらい *turai* 'tough' and そｒ for それ *sore* 'it.'

5.5. Experimental Settings

5.5.1. Corpus Statistics

We present the statistics of BQNC in Table 5.4. It comprises 929 sentences, 12.6K word tokens, and 767 non-standard word tokens. As shown in Table 5.6, the corpus contains tokens of seven types of vocabulary and four types of variant

[50] We assessed whether a form is unusual if its occurrence rate was approximately equal to 5% or less in BCCWJ similar to the case of character type variants.

Register	# sent	# word token	# word type	# NSW token	# NSW type
Q&A	379	5,649	1,699	320	221
Blog	550	6,951	2,231	447	257
Total	929	12,600	3,419	767	420

Table 5.4. Statistics of BQNC. NSW represents non-standard word.

form. Whereas there exist fewer than 40 instances of neologisms/slang, dialect words, foreign words, and typographical errors, each of the other category has more than 100 instances. Our corpus contains a similar number of non-standard tokens to Kaji and Kitsuregawa's [51] Twitter corpus (1,831 sentences, 14.3K tokens, and 793 non-standard tokens) and Osaki et al.'s [96] Twitter corpus (1,405 sentences, 19.2K tokens, and 768 non-standard tokens). The former follows the POS tags for the Japanese MA toolkit JUMAN and the latter follows the authors own POS tags that extend NINJAL's SUW.

5.5.2. Systems

We evaluated two existing methods for MA and lexical normalization on BQNC. First, we used MeCab 0.996[51] [57], which is a popular Japanese MA toolkit based on CRFs. We used UniDic[52] (unidic-cwj-2.3.0) as the analysis dictionary, which contains attribute information of 873K words and MeCab's parameters (word occurrence costs and transition costs) learned from annotated corpora, including BCCWJ [23].

Second, we used our implementation of Sasano et al.'s [112] joint MA and normalization method. They defined derivation rules to add new nodes in the word lattice of an input sentence built by their baseline system, JUMAN. Specifically, they used the following rules: (i) sequential voicing (*rendaku*), (ii) substitution with long sound symbols and lowercase kana, (iii) insertion of long sound symbols and lowercase kana, (iv) repetitive onomatopoeia (XYXY-form[53]) and (v) non-

[51] `https://taku910.github.io/mecab/`

[52] `https://unidic.ninjal.ac.jp/`

[53] "X" and "Y" represent the same kana character(s) corresponding to one mora, "Q" represents a long consonant character "っ"/"ッ," "*ri*" represents a character "り"/"リ," and "*to*" represents

Task	MeCab			MeCab+ER		
	P	R	F_1	P	R	F_1
Seg	89.2	95.1	92.1	**93.5**	**96.5**	**95.0**
POS	87.5	93.3	90.3	**91.4**	**94.3**	**92.8**
Norm	–	–	–	55.9	25.8	35.3

Table 5.5. Precision (P), Recall (R), and F_1 score for for the three tasks: segmentation (Seg), POS tagging, and normalization (Norm).

repetitive onomatopoeia (XQY*ri*-form and XXQ*to*-form). For example, rule (iii) adds a node of 冷たぁぁい *tsumetāi* as a variant form of 冷たい *tsumetai* 'cold' and rule (iv) adds a node of うはうは *uhauha* 'exhilarated' as an onomatopoeic adverb if the input sentences contain such character sequences.

The original implementation by Sasano et al. [112] was an extension of JUMAN and followed the JUMAN POS tag set. To adapt their approach to SUW, we implemented their rules and used them to extend the first method of MeCab using UniDic. We set the costs of the new nodes by copying the costs of their standard forms or the most frequent costs of the same-form onomatopoeia, whereas Sasano et al. [112] manually defined the costs of each type of new word. We denote this method by MeCab+ER (Extension Rules). Notably, we did not conduct any additional training to update the models' parameters for either methods.

5.6. Results and Analysis

5.6.1. Overall Results

Table 5.5 shows the overall performance, that is, precision, recall, and F_1 score of both methods for segmentation, POS tagging and normalization.[54] Compared with well-formed text domains,[55] the relatively lower performance (F_1 of 90–95%) of both methods for segmentation and POS tagging indicates the difficulty of ac-

a character "と"/"ト."

[54] We only evaluated top-level POS for POS tagging. We regarded a predicted standard form as correct if the prediction was equal to one of the gold standard forms for normalization.

[55] For example, Kudo et al. [57] achieved F_1 of 98%–99% for segmentation and POS tagging in news domains.

Category	No.	MeCab		MeCab+ER	
		Seg	POS	Seg	POS
Dialect words	23	91.3	78.3	**95.7**	**82.6**
Proper names	103	87.4	84.5	**88.4**	**85.4**
Onomatopoeia	218	79.8	73.4	**87.2**	**77.1**
Foreign words	14	78.6	78.6	78.6	78.6
Emoticons/AA	270	73.7	64.1	73.3	63.3
Interjections	174	64.9	**53.5**	**72.4**	48.9
Neologisms/Slang	37	67.6	67.6	67.6	67.6
Sound change variants	419	50.6	47.5	**82.6**	**76.4**
Char type variants	248	71.0	62.9	**78.2**	**69.4**
Alternative representations	132	65.2	54.6	**76.5**	**69.0**
Typographical errors	23	47.8	30.4	47.8	30.4
Non-general/standard total	1,565	68.9	61.9	79.6	70.4
Standard forms of general words	11K	98.9	97.7	98.9	97.7

Table 5.6. Recall for each category (Segmentation and POS tagging).

curate segmentation and tagging in UGT. However, MeCab+ER outperformed MeCab by 2.5–2.9 F_1 points because of the derivation rules. Regarding the normalization performance of MeCab+ER, the method achieved moderate precision but low recall, which indicates its limited coverage for various variant forms in the dataset.

5.6.2. Results for Each Category

Table 5.6 shows segmentation and POS tagging recall of both methods for each category. In contrast to the sufficiently high performance for general words, both methods performed worse for words of characteristic categories in UGT; micro average recall was at most 79.6% for segmentation and 70.4% for POS tagging ("non-general/standard total" column). MeCab+ER outperformed MeCab particularly for onomatopoeia, character type variants, alternative representations, and sound change variants. The high scores for dialect words were probably because UniDic contains a large portion of (19 out of 23) dialect word tokens. Interjection was a particularly difficult vocabulary type, for which both methods

Category	No.	MeCab+ER
Sound change variants	419	37.0
Character type variants	248	0.0
Alternative representations	132	32.6
Typographical errors	23	0.0

Table 5.7. Recall for each category (Normalization).

MeCab\MeCab+ER	T	F
T	11,955	32
F	200	413

Table 5.8. The number of correct (T) or incorrect (F) segmentation for two methods.

recognized only approximately 50% of the gold POS tags. We guess that this is because the lexical variations of interjections are diverse; for example, there are many user-generated expressions that imitate various human voices, such as laughing, crying, and screaming.

Table 5.7 shows the recall of MeCab+ER's normalization for each category. The method correctly normalized tokens of alternative representations and sound change variants with 30–40% recall. However, it completely failed to normalize character type variants not covered by the derivation rules and more irregular typographical errors.

5.6.3. Analysis of Segmentation Results

We performed error analysis of the segmentation results for the two methods. Table 5.8 shows a matrix of the number of correct or incorrect segmentation of the methods for gold words. There existed 32 tokens that only MeCab correctly segmented (T-F), 200 tokens that only MeCab+ER correctly segmented (F-T), and 413 tokens that both methods incorrectly segmented (F-F).

In Table 5.9, we show the actual segmentation/normalization examples using the methods for the three cases; the first, second, and third blocks show examples of T-F, F-T, and F-F cases, respectively. First, out of 32 T-F cases, MeCab+ER incorrectly segmented tokens as onomatopoeia in 18 cases. For example, (a) and (b) correspond to new nodes added by the rules for the XQY*ri*-form and XYXY-

	VT	Gold Seg&SForms	Reading	Translation	MeCab result	MeCab+ER result
(a)		はっ|たり	*haQ*|*tari*	paste and	はっ|たり	はったり
(b)		こら|こら	*kora*|*kora*	hey hey	こら|こら	こらこら
(c)	S	しーかーも [しかも]	*shīkāmo*	besides	しー|かー|も	しーかーも [しかも]
(d)	A	ぉぃら [おいら,オイラ]	*oira*	I	ぉ|ぃ|ら	ぉぃら [おいら]
(e)	S	んまぃ [美味い,旨い,うまい]	*mmai*	yummy	ん|ま|ぃ	んまぃ [んまい]
(f)	C,A	も|やきゅー [野球]	*mo*|*yakyū*	also, baseball	もや|きゅー	も|やきゅー [やきゅう]
(g)	S	たしーか [確か,たしか] |に	*tashīka*|*ni*	surely	た|し|ー|かに	たしーか [たしか] |に
(h)		ふぅ〜〜ん	*fūn*	hmm	ふぅ〜|〜|ん	ふぅ〜〜ん [ふん]
(i)	S	ませう〜 [ましょう]	*mashō*	let's	ませ|う|〜	ませ|う〜 [う]
(j)	C,S	けこーん [結婚]	*kekōn*	marriage	け|こーん	け|こー [こう] |ん
(k)	A	ください|n e [ね]	*kudasai*|*ne*	Won't you...?	ください|n|e	ください|n|e
(l)		（＾へ＾）			（|＾|へ|＾|）	（|＾|へ|＾|）
(m)		社割	*shawari*	employee discount	社|割	社|割
(n)		ガルバディア	*garubadhia*	Galbadia	ガルバ|ディア	ガルバ|ディア

Table 5.9. Segmentation and normalization results (shown in "[]") by MeCab and MeCab+ER. Incorrect results are written in gray. VT represents variant type. C, A, and S represent character type variant, alternative representation, and sound change variants, respectively. Gold Seg&SForms represent the gold segmentation and gold standard forms (shown in "[]").

form onomatopoeia, respectively, even though (a) is a verb phrase and (b) is a repetition of interjections.

Second, out of 200 F-T cases that only MeCab+ER correctly segmented, the method correctly normalized 119 cases, such as (c), (d), and the first word たしーか in (g), and incorrectly normalized 42 cases, such as (e) and the second word やきゅー in (f). The remaining 39 cases were tokens that required no normalization, such as the first word も in (f), the second word に in (g), and (h). The method correctly normalized simple examples of sound change variants (c: しーかーも for しかも) and alternative representations (d: ぉぃら for おいら) because of the substitution and insertion rules, but failed to normalize character type variants (f: やきゅー for 野球) and complicated sound change variants (e: んまぃ for うまい).

Third, out of 413 F-F cases, 148 tokens were complicated variant forms, including a combination of historical kana orthography and the insertion of the long sound symbol (i), a combination of the character type variant and sound change variant (j), a variant written in *romaji*, namely, Roman letter transcription (k).

Total		T-SEG				F-SEG
Gold	767	TP	198	FN	58	511
Pred	354	TP	198	FP	99	57

Table 5.10. Detailed normalization results by MeCab+ER.

The remaining 265 tokens were other unknown words, including emoticons (l), neologisms/slang (m), and proper names (n).[56]

5.6.4. Analysis of Normalization Results

Table 5.10 shows the detailed normalization results for MeCab+ER. Among 767 non-standard words (Gold), the method correctly normalized 198 TPs and missed 569 (58+511) FNs. Similarly, among 354 predictions (Pred), the methods incorrectly normalized 156 (99+57) false positives (FP). We further divided FN and FP examples according to whether they were correctly segmented (T-SEG) or not (F-SEG).

We do not show TP and FN examples here since we already introduced some examples in §5.6.3. Among the FP examples, some of them were not necessarily inappropriate results; normalization between similar interjections and onomatopoeia was intuitively acceptable (e.g., おお〜 was normalized to おお *ō* 'oh' and サラサラ〜 was normalized to サラサラ *sarasara* 'smoothly'). However, we assessed these as errors based on our criterion that interjections had no (non-)standard forms and the BCCWJ guidelines that regarded onomatopoeia with and without long sound insertion as different lemmas.

5.6.5. Discussion

The derivation rules used in MeCab+ER improved segmentation and POS tagging performance and contributed to the correct normalization of parts of variant forms, but the overall normalization performance was limited to F_1 of 35.3%.

We classified the main segmentation and normalization errors into two types: complicated variant forms and unknown words of specific vocabulary types such

[56] 社割 *shawari* is an abbreviation of 社員割引 *shain waribiki* 'employee discount.' ガルバディア 'Galbadia' is an imaginary location name in the video game Final Fantasy.

as emoticons and neologisms/slang. The effective use of linguistic resources may be required to build more accurate systems, for example, discovering variant form candidates from large raw text similarly to Saito et al. [107], and constructing/using term dictionaries of specific vocabulary types.

5.7. Related Work

UGT Corpus for MA and Normalization. Hashimoto et al. [37] developed a Japanese blog corpus with morphological, grammatical, and sentiment information, but it contains only 38 non-standard forms and 102 misspellings as UGT-specific examples. Osaki et al. [96] constructed a Japanese Twitter corpus annotated with morphological information and standard word forms. Although they published tweet URLs along with annotation information,[57] we could only restore parts of sentences because of the deletion of the original tweets. Some previous work [51, 107, 108, 112] developed Japanese MA and lexical normalization methods for UGT, but most of their in-house data are not publicly available.

For English lexical normalization, Han and Baldwin [35] constructed an English Twitter corpus and Yang and Eisenstein [159] revised it as LexNorm 1.2. Baldwin et al. [7] constructed an English Twitter corpus (LexNorm2015) for the W-NUT 2015 text normalization shared task. Both LexNorm 1.2 and LexNorm2015 have been used as benchmark datasets for normalization systems [22, 49, 138].

For Chinese, Li and Yarowsky [61] published a dataset of formal-informal word pairs collected from Chinese webpages. Wang et al. [141] released a crowdsourced corpus constructed from microblog posts on Sina Weibo.

Classification of Linguistic Phenomena in UGT. To construct an MA dictionary, Nakamoto et al. [91] classified unknown words occurring in Japanese chat text into contraction (e.g., すげー for すごい *sugoi* ‘awesome’), exceptional kana variant (e.g., こんぴゅーた for コンピュータ ‘computer’), abbreviation (e.g., メアド for メールアドレス ‘mail address’), typographical errors, filler, phonomime and phenomime, proper nouns, and other types. Ikeda et al. [46] classified “peculiar expressions” in Japanese blogs into visual substitution (e.g.,

[57] https://github.com/tmu-nlp/TwitterCorpus

わたU for わたし *watashi* ‘me’), sound change (e.g., でっかい for でかい *dekai* ‘big’), kana substitution (e.g., びたみん for ビタミン ‘vitamin’), and other unknown words into similar categories to Nakamoto et al. [91]. Kaji et al. [52] performed error analysis of Japanese MA methods on Twitter text. They classified mis-segmented words into a dozen categories, including spoken or dialect words, onomatopoeia, interjections, emoticons/AA, proper nouns, foreign words, misspelled words, and other non-standard word variants. Ikeda et al.’s [46] classification of peculiar expressions is most similar to our types of variant forms and Kaji et al.’s [52] classification is most similar to our types of vocabulary (shown in Table 5.2), whereas we provide more detailed definitions of categories and criteria for standard and non-standard forms. Other work on Japanese MA and lexical normalization did not consider diverse phenomena in UGT [108, 112].

For English, Han and Baldwin [35] classified ill-formed English words on Twitter into extra/missing letters and/or number substitution (e.g., “b4” for “before”), slang (e.g., “lol” for “laugh out loud”), and “others.” van der Goot et al. [139] defined a more comprehensive taxonomy with 14 categories for a detailed evaluation of English lexical normalization systems. It includes phrasal abbreviation (e.g., “idk” for “I don’t know”), repetition (e.g., “soooo” for “so”), and phonetic transformation (e.g., “hackd” for “hacked”).

For Chinese, Li and Yarowsky [61] classified informal words in Chinese webpages into four types: homophone (informal words with similar pronunciation to formal words, e.g., 稀饭 ⟨xīfàn⟩[58] ‘rice gruel’ for 喜欢 ⟨xǐhuan⟩ ‘like’), abbreviation and acronym (e.g., GG for 哥哥 ⟨gēge⟩ ‘elder brother’), transliteration (informal words are transliteration of English translation of formal words, e.g., 3Q ⟨sānqiu⟩ for 谢谢 ⟨xièxie⟩ ‘thank you’), and “others.” Wang et al. [141] also classified informal words in Chinese microblog posts similar to Li and Yarowsky [61].

5.8. Conclusion

We presented our publicly available Japanese UGT corpus annotated with morphological, normalization, and word category information. Our corpus enables

[58] Pinyin pronunciation is shown in “⟨⟩”.

the performance comparison of existing and future systems and identifies the main remaining issues of MA and normalization of UGT. Experiments on our corpus demonstrated the limited performance of the existing systems for non-general words and non-standard forms mainly caused by two types of difficult examples: complicated variant forms and unknown words of non-general vocabulary types.

6. A Text Editing Approach to Joint Japanese Word Segmentation, POS Tagging, and Lexical Normalization

6.1. Introduction

UGT is a valuable source of knowledge and opinions from diverse users. A notable characteristic of UGT is that it contains non-canonical sentences, and this degrades the performance of NLP systems trained on canonical sentences. To reduce the gap between the performance on general text and on UGT, lexical normalization techniques have been explored, particularly for English [5, 7]. In addition, Japanese UGT requires a further step: to identify nonstandard words in unsegmented sentences written without word delimiters. For this reason, the problem of Japanese lexical normalization has been solved by predicting word boundaries, POS tags, and normalized word forms simultaneously [108, 112]. Similarly to previous work, we tackle the joint task comprising Japanese word Segmentation, POS tagging, and lexical Normalization (SPN).

A critical problem in lexical normalization is the lack of labeled data. Manual annotation of normalized forms is a time-consuming task; therefore, the size of available annotated corpora is quite small. A prospective solution to this problem is the use of pseudo-labeled data. In this chapter, we propose methods of generating pseudo-labeled data using (auto-) segmented sentences and standard and nonstandard word variant pairs. To generate high quality labels, we acquire reliable variant pairs based on lexical knowledge, namely, a dictionary with lemma definition and hand-crafted rules.

For efficient learning from a limited amount of data, we adopt a text editing approach. Our neural tagging model predicts edit operations to normalize input characters, while predicting segmentation and POS tags at the same time. The editing process is similar to that proposed in previous work on English lexical normalization [18, 78], but we design a specific tag set for the Japanese SPN task, which requires the management of a large number of character types.

Our extensive experiments on the SPN task demonstrated that our model achieved better normalization performance when the model used more additional features, it was trained on more types of pseudo-labeled data, and it was trained on training instances with more diverse context.

6.2. Proposed Method

6.2.1. Multiple Sequence Labeling Formulation

In this work, we formulate the SPN task as multiple character-level sequence labeling problems. We convert the label sequence $\boldsymbol{t}$, defined in §2.2.1, to four tag sequences: a segmentation tag sequence $\boldsymbol{t}^s$, a character-level POS tag sequence $\boldsymbol{t}^p$, a string edit operation (SEdit) tag sequence $\boldsymbol{t}^e$, and a character type conversion (CConv) tag sequence $\boldsymbol{t}^c$.

We employ a tag set $\mathcal{T}_{\text{seg}} = \{\texttt{B}, \texttt{I}, \texttt{E}, \texttt{S}\}$ for segmentation, which is described in §2.1.1. We set $t_i^p = p_j \in \mathcal{T}_{\text{pos}}$ for the POS tag of a character x_i in a word w_j $(f_j \leq i \leq l_j)$, where $\mathcal{T}_{\text{pos}}$ denotes a POS tag set. We use two types of tags for the normalization task. For x_i in a standard word w_j, we set $t_i^e = t_i^c = \texttt{KEEP}$, which means that no edit operation or conversion is required for x_i. For x_i in a nonstandard word w_j, two types of tags $t_i^e \in \mathcal{T}_{\text{sedit}}$ and $t_i^c \in \mathcal{T}_{\text{cconv}}$ are generated based on the closest standard form $s_j^\star \in S_j$, where $\mathcal{T}_{\text{sedit}}$ and $\mathcal{T}_{\text{cconv}}$ represent the tag sets of SEdit and CConv, which we define in §6.2.2. The procedure for selecting the closest standard form is as follows: a character alignment between w_j and $s \in S_j$ is calculated, and then the standard form with the most characters aligned to w_j is selected.[59]

[59] We describe the procedure in detail in Appendix §D.

	Meaning	Nonstandard word w	Standard form s	SEdit tags $\boldsymbol{t}^e$	CConv tags $\boldsymbol{t}^c$
(a)	really	まぢ	まじ	K, REP(じ)	K, K
(b)	difficult	ムズカシー	むずかしい	K, K, K, K, REP(い)	HR, HR, HR, HR, K
(c)	terrific	すごーいー	すごい	K, K, D, K, K, D	K, K, K, K, K
(d)	high/expensive	たっけぇ	たかい	K, REP(か), REP(い), D	K, K, K, K
(e)	awesome	さいこー	最高	K, K, K, REP(う)	KJ, KJ, KJ, KJ

Table 6.1. Examples of labels for nonstandard and standard word pairs. K, D, HR, and KJ represent KEEP, DEL, TO_HIRAGANA, and TO_KANJI, respectively.

6.2.2. Tag Definition

The Japanese writing system comprises three major scripts: hiragana, katakana, and kanji. The numbers of character types in them are different: approximately 80 in hiragana, 80 in katakana,[60] and more than 4,000 in kanji. To decrease the tag space size, we allow insertion and replacement operations only for kana characters $\mathcal{V}_{\text{kana}}$. Specifically, we define the SEdit tags as $\mathcal{T}_{\text{sedit}} = \{\texttt{KEEP}, \texttt{DEL}, \texttt{INSL(}c\texttt{)}, \texttt{INSR(}c\texttt{)}, \texttt{REP(}c\texttt{)}\}$ for $c \in \mathcal{V}_{\text{kana}}$. DEL indicates deletion of the current character, INSL(c) and INSR(c) indicate insertion of c immediately to the left and right of the current character, respectively, and REP(c) indicates replacement of the current character by c. In addition, we define the CConv tags as $\mathcal{T}_{\text{cconv}} = \{\texttt{KEEP}, \texttt{TO\_HIRA}, \texttt{TO\_KATA}, \texttt{TO\_KANJI}\}$, where the last three tags indicate conversion of the current character to hiragana, katakana, and kanji, respectively.[61]

For the example sentence $\boldsymbol{x}$ =“日本|語|まぢ|ムズカシー” with the standard forms $S_3 = \{$まじ,マジ$\}$ and $S_4 = \{$難しい,むずかしい$\}$ in Table 2.3, the tags for $w_3 = \boldsymbol{x}_{4:5}$ =まぢ and $w_4 = \boldsymbol{x}_{6:10}$ =ムズカシー are shown as (a) and (b) in Table 6.1, and the tags for the other characters are $t^e_i = t^c_i = \texttt{KEEP}$ $(1 \leq i \leq 3)$. Both types of tags are automatically generated, according to the character alignments between original and standard tokens. Table 6.1 lists examples (c)–(e), which have other types of tags.

A remaining problem is the ambiguity of characters assigned with the TO_KANJI

[60] We distinguish kana characters with and without a voicing mark (e.g., “か” *ka* and “が” *ga*).

[61] Tag definition different from above could be used. We investigated two alternative settings, but our preliminary experiments showed no gains over our proposed setting: a case where SEdit and CConv tags were merged into a single tag set and a case where additional SEdit tags similar to the special operators used in the pronunciation feature (§6.2.3) were introduced.

tag; for example, あき *aki* can be converted to 秋 'autumn', 空き 'vacancy', or 飽き 'bored' depending on its surrounding context. We use an external kana-to-kanji converter to select the most likely candidates.

There still exist cases in which a nonstandard word with many deleted or replaced characters cannot be restored to its standard form (e.g., よろ *yoro* to よろしく *yoroshiku* 'thank you') by the defined tags when the required number of insertion and replacement operations exceeds the number of characters in the original token. This can be solved by introducing multi-character operations (e.g., `INSR(しく)`), but we assume that most instances can be expressed by single-character operations, and leave those cases for future work.

6.2.3. Model Architecture

For the sequence labeling tasks, we use a BiLSTM-based model that consists of BiLSTM layers and task-specific softmax layers,[62] which extends the baseline single-task BiLSTM model in §2.3. An input character sequence $\boldsymbol{x}$ is transformed to embedding vectors $\boldsymbol{e}_{1:n} = (\boldsymbol{e}_1, \cdots, \boldsymbol{e}_n)$ and fed into multi-layer BiLSTM. Hidden vectors from forward and backward LSTM are concatenated, to form a single hidden vector $\boldsymbol{h}_i$ for each character. $\boldsymbol{h}_i$ is then mapped to a score distribution vector $\boldsymbol{y}_i^u$ via a softmax layer for each task $u \in \{s, p, e, c\}$, each of which indicates segmentation, POS tagging, SEdit, and CConv task, respectively.

Given training data $\mathcal{D}$, the model parameters are learned by minimizing a loss function L during training. The loss L is defined as the sum of the cross-entropy between the gold and predicted tag distributions for all tasks:

$$L = -\sum_{\boldsymbol{x}\in\mathcal{D}} \sum_{u\in\{s,p,e,c\}} \lambda_u \sum_{1\leq i\leq|\boldsymbol{x}|} \boldsymbol{t}_i^u \log\ \boldsymbol{y}_i^u\ , \tag{6.1}$$

where λ_u is a coefficient to control the contribution of each task u and $\boldsymbol{t}_i^u$ is the one-hot vector of the gold label t_i^u.

[62] We adopt the softmax-based prediction for efficiency because the number of SEdit tags was about 250 in our experiments.

6.2.4. Features

We use two input features based on pronunciation and lexicon entries, in addition to character embedding features described in §2.3. Feature vectors from the three sources for each character are concatenated, to form a single vector $\boldsymbol{e}_i$ in §6.2.3.

Pronunciation Feature We introduce a pronunciation element that corresponds to a vowel, a consonant, the long sound symbol, or a special operator (voicing V, semi-voicing P, or lowercasing S) in a kana character sequence.[63] These elements are similar to romaji (Roman letter transcription) but differ mainly with respect to the special operators. For example, "グ" *gu*, "ァ" *a*, and "パ" *pa* are decomposed into $\{k, u, V\}$, $\{a, S\}$, and $\{h, a, P\}$, respectively. Each character x_i is decomposed into one or more pronunciation elements. A pronunciation vector $\boldsymbol{e}_i^p$ for x_i is the average of its pronunciation element embeddings retrieved from an embedding matrix.

Lexicon Feature We define two types of binary features based on a nonstandard word lexicon.[64] A lexicon word feature for a character x_i is defined as a $(|P| \times |K|)$-dimensional vector $\boldsymbol{e}_i^{d,w}$, each element of which indicates whether x_i corresponds to a particular position $p \in P = \{$immediate left, immediate right, beginning, middle, end$\}$ of any nonstandard word of length $k \in K$ in the lexicon. Similarly, a lexicon POS feature for x_i is defined as a $|\mathcal{T}_{\text{POS}}|$-dimensional vector $\boldsymbol{e}_i^{d,p}$, each element of which indicates whether the x_i corresponds to an inside position of any nonstandard word with a particular POS.

6.3. Pseudo-labeled Data Generation

To overcome the lack of training data for the normalization task, we construct a set of standard and nonstandard word variant pairs $\mathcal{V}$ and then generate different

[63] We generate pronunciation features only for kana characters and use zero vectors for other types of characters.

[64] We use nonstandard words in dictionary-derived ($\mathcal{V}_d$) and rule-derived variant pairs ($\mathcal{V}_r$) (described in §6.3) for the models trained on dictionary-derived ($\mathcal{A}_d$) and rule-derived data ($\mathcal{A}_r$) (described in §6.4.1) in our experiments, respectively.

Original sentence $\boldsymbol{x}_a$				Updated information		
Word	Lemma ID	POS	Conjugation form	SEdit tags	CConv tags	Standard form
スゴく	19163	Adjective	Continuative	K, K, K	HIRA, HIRA, K	すごく
気	8263	Noun	–	K	K	–
に	28178	Particle	–	K	K	–
なる	28061	Verb	Termination	K, K	K, K	–

Table 6.2. An example of $\mathrm{DS}_{\mathrm{tgt}}$. A segmented sentence $\boldsymbol{x}_a$ is annotated with SEdit and CConv tags by DS, according to the matched token スゴく with the same lemma ID and conjugation form as those of p_a =(スゴく, すごく).

Original sentence $\boldsymbol{x}_b$				Updated information			
Word	Lemma ID	POS	Conjugation form	Word (Replaced)	SEdit tags	CConv tags	Standard form
ほんとう	34947	Noun	–	ほんっと	K, K, D, INSR(う)	K, K, K, K	ほんとう
に	28198	Particle	–	–	K	K	–
心配	19516	Noun	–	–	K, K	K, K	–
です	25653	Copula	Termination	–	K, K	K, K	–

Table 6.3. An example of $\mathrm{DS}_{\mathrm{src}}$. For variant pair p_b, a segmented sentence x_b containing a token with the same lemma ID and conjugation form as those of p_b is extracted. Then, a synthetic sentence "ほんっと|に|心配|です" annotated with SEdit and CConv tags is generated by $\mathrm{DS}_{\mathrm{src}}$.

types of pseudo-labeled data by two approaches: distant supervision on formal target-side ($\mathrm{DS}_{\mathrm{tgt}}$) and informal source-side text ($\mathrm{DS}_{\mathrm{src}}$).

$\mathrm{DS}_{\mathrm{tgt}}$ generate a sentence where the original tokens are retained but nonstandard tokens among them are annotated with pseudo standard tokens; specifically, given a segmented sentence, a token matching with a nonstandard form v_{nst} in $\mathcal{V}$ is annotated with SEdit and CConv tags to convert to its standard form, while other tokens are annotated with KEEP tags. Table 6.2 shows an example of $\mathrm{DS}_{\mathrm{tgt}}$ to a segmented sentence $\boldsymbol{x}_a$ ="スゴく|気|に|なる" *sugoku ki ni naru* '(I'm) very curious' and a standard and nonstandard word variant pair p_a =(スゴく, すごく) 'very' with lemma ID of 19163 and continuative form.

On the other hand, $\mathrm{DS}_{\mathrm{src}}$ generate a sentence where one or more of the original tokens are replaced by pseudo nonstandard forms; given a standard and nonstandard variant pair $(v_{\mathrm{st}}, v_{\mathrm{nst}}) \in \mathcal{V}$, $\mathrm{DS}_{\mathrm{src}}$ extracts a segmented sentence containing a token with the same lemma as that of the pair, replaces the token by v_{nst},

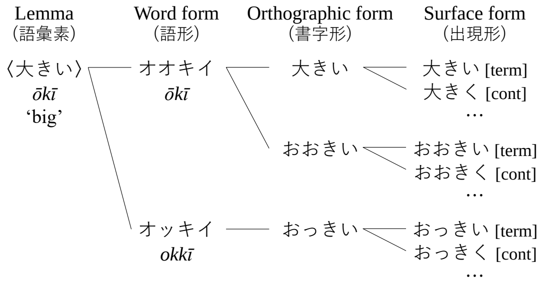


Figure 6.1. Hierarchical lemma definition in UniDic. Termination forms (term) and continuative forms (cont) are illustrated as examples of surface forms.

and generates SEdit and CConv tags to convert v_{nst} to v_{st}. Table 6.3 shows an example of DS$_{\mathrm{src}}$ to a segmented sentence $\boldsymbol{x}_b$ =“ほんとう|に|心配|です” *hontō ni shimpai desu* ‘(I'm) really worried’ and a variant pair p_b =(ほんとう, ほんっと) ‘truth’ with lemma ID of 34947.

As the main difference between two approaches, DS$_{\mathrm{tgt}}$ does not change original sentences but DS$_{\mathrm{src}}$ does. Although we can use actual sentences with DS$_{\mathrm{tgt}}$, we can easily obtain any number of synthetic sentences containing nonstandard words of interest with DS$_{\mathrm{tgt}}$. However, both approaches require reliable variant pairs to generate useful data. For this purpose, we use two strategies for variant pair acquisition: dictionary-based and rule-based.

6.3.1. Dictionary-derived Variant Pairs

As the first approach to variant pair acquisition, we use UniDic (unidic-cwj-2.3.0), which employs hierarchical definition of word indexes, but any dictionary with lemma and conjugation information can be used. As shown in Figure 6.1, a lemma in UniDic aggregates word forms with different pronunciation and word forms with different conjugation types, a word form aggregates orthographic forms, and an orthographic form aggregates surface forms. Thus, surface forms with the same lemma and conjugation form compose a variant set; for example, the continuative surface forms of a lemma ⟨大きい⟩ include 大きく *ōkiku*, おおきく *ōkiku*, and おっきく *okkiku*.

We extract valid standard and nonstandard word pairs from variant sets by the following steps. (1) Words whose POS is symbol, space, person name, or number are excluded. (2) Each variant in a variant set is automatically classified as a standard form or valid nonstandard form by predefined rules,[65] which are based on pronunciation and frequency of occurrence among the variant forms of the lemma in a corpus. (3) Finally, each nonstandard form is paired with the closest standard form.

6.3.2. Rule-derived Variant Pairs

As an alternative approach, we use hand-crafted rules to transform standard forms to nonstandard forms. We classify lexical variations that have been reported in previous work [52, 81] or observed by us into dozens of patterns. We then choose patterns that are easy-to-implement or widely adaptable to many words, and implement them as variant generation rules. Specifically, we use four rules: change of character type (e.g., 疲労 *hirō* 'fatigue' → ひろう), and substitution by a character with the same pronunciation (e.g., マジ *maji* 'really' → マヂ), mora consonant (e.g., 行こう *ikō* 'go' → 行こっ), and uppercase kana (e.g., ちょっと *chotto* 'bit' → ちよつと), as well as six rules similar to those used by Sasano et al. [112] and Ikeda et al. [47]. We describe the rules in detail in Appendix §E

To obtain plausible variant pairs, we follow these steps: (1) apply the rules to standard forms, which are obtained by the dictionary-based approach, and generate nonstandard form candidates, (2) count frequencies of character n-grams that match original standard forms or generated nonstandard forms in a corpus, and (3) accept variant pairs of which both the standard and nonstandard forms have frequencies higher than a threshold value of 10.

6.4. Experimental Settings

6.4.1. Language Resources

As training data $\mathcal{D}_t$ and development data $\mathcal{D}_v$ for the segmentation and POS tagging tasks, we used 57K and 3K sentences, respectively, with the SUW annotation

[65] The classification procedure is as follows: a variant with different pronunciation from its lemma is regarded as a nonstandard form, a variant with a frequency of occurrence of 5% or less as a nonstandard form, and a variant with a frequency of 10% or more as a standard form.

from the BCCWJ core data.

For variant pair acquisition, we counted the frequencies of UniDic entries (§6.3.1) using 3.5M sentences in the BCCWJ non-core data and character n-gram frequencies (§6.3.2) using 8.8M sentences in Yahoo! Chiebukuro data.[66]

We constructed three pseudo-labeled datasets using the dictionary-derived and rule-derived variant pairs, which we denote by $\mathcal{V}_d$ and $\mathcal{V}_r$. The first dataset $\mathcal{A}_t$ was generated by applying $\mathrm{DS}_{\mathrm{tgt}}$ to $\mathcal{D}_t$ using $\mathcal{V}_d$.[67] The second dataset $\mathcal{A}_d$ and third dataset $\mathcal{A}_r$ are 173K and 170K synthetic sentences generated by $\mathrm{DS}_{\mathrm{src}}$ from the 3.5M BCCWJ non-core sentences, using the top $n_p = 20$K frequent pairs[68] in $\mathcal{V}_d$ or $\mathcal{V}_r$. Notably, for each pair in $\mathcal{V}_d$ or $\mathcal{V}_r$, we extracted at most $n_s = 10$ original sentences that contained their lemma.[69] Similarly, we constructed an additional 3K sentences from the top 3K frequent pairs in $\mathcal{V}_d$ and 3K from 3K pairs in $\mathcal{V}_r$, and used them as development data, together with $\mathcal{D}_v$.

As test data, we used BQNC described in Chapter 5.

6.4.2. Training Setting

During each training epoch, we randomly constructed a mini-batch consisting only of real training sentences ($\mathcal{A}_t$) or synthetic training sentences ($\mathcal{A}_d$ or $\mathcal{A}_r$) for each iteration, and trained the model for up to 20 epochs. We randomly initialized all parameters, applied mini-batch stochastic gradient descent to optimize parameters, and reduced the learning rate by a fixed decay ratio every epoch after the first five epochs. We set the loss coefficient values in Eq. (6.1) as $(\lambda_s, \lambda_p, \lambda_e, \lambda_c) = (1, 1, 1, 1)$ for real sentences, and set them as $(\lambda_s, \lambda_p, \lambda_e, \lambda_c) = (\lambda_0, \lambda_0, 1, 1)$ for synthetic sentences with automatic segmentation and POS tags, where λ_0 is a hyperparameter.

We searched for the best values, within given ranges, for some hyperparameters of the model and used predetermined values for the others: character embedding size 200 from {100, 200, 300}, pronunciation embedding size 30 from {10, 20, 30,

[66] `https://www.nii.ac.jp/dsc/idr/yahoo/chiebkr3/Y_chiebukuro.html`

[67] We did not construct $\mathrm{DS}_{\mathrm{tgt}}$-based data using $\mathcal{V}_r$.

[68] We obtained 404K pairs from 873K UniDic entries and 47K pairs from 868K rule-generated nonstandard form candidates by the process described in §6.3.1 and §6.3.2.

[69] Fewer than 200K sentences were generated because 10 sentences were not necessarily included for every variant pair in the original corpus.

40, 50}, BiLSTM hidden unit size 1,000 from {200, 400, 600, 800, 1,000}, BiLSTM dropout [161] rate 0.1 from {0.1, 0.2, 0.3, 0.4}, loss coefficient $\lambda_0 = 0.4$ from {0.2, 0.4, 0.6, 0.8, 1.0}, number of BiLSTM layers 2, mini-batch size 100, initial learning rate 1.0, learning rate decay ratio 0.9, and gradient clipping threshold 5.0.

6.4.3. Kana-to-Kanji Conversion Model

For kanji conversion (KC), we trained an n-gram language model (LM) of kana-kanji mixed sentences using SRILM[70] [122] on 1.2M sentences in the BCCWJ core and non-core data, and performed Viterbi decoding with negative log probability of the LM using the `decode_ngram.py`[71] script. Specifically, if the `TO_KANJI` tag was predicted for more than one character in a predicted word span by the normalization model, the word was given to the KC model, together with six preceding and six succeeding characters, and the best hypothesis found was output as a normalized form.

6.4.4. Post-processing

We defined post-processing rules to apply to the predicted segmentation or normalization results. The first rule SP changes segmentation tags $y^s_{i+1:i+k}$ to $\texttt{I} \in \mathcal{T}_{\text{seg}}$ when k consecutive characters $x_{i:i+k}$ are the same vowel kana, long sound symbol, mora nasal, or mora consonant characters, according to our finding that such cases were rare from our preliminary experiment. The second rule NP changes a predicted normalized word to its original string if the predicted form is not in a standard form lexicon, which corresponds to the standard forms in $\mathcal{V}_d$.

6.4.5. Baseline Methods

We evaluated the two systems described in §5.5.2 for comparison: MeCab [57] 0.996 with UniDic (unidic-cwj-2.3.0) and our implementation of Sasano et al.'s [112] joint MA and normalization method, which we call MeCab+ER.

[70] `http://www.speech.sri.com/projects/srilm/`

[71] `https://github.com/yohokuno/neural_ime`

Method	Use of data			Seg			POS			Norm		
	$\mathcal{A}_t$	$\mathcal{A}_r$	$\mathcal{A}_d$	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$
MeCab				89.2	95.1	92.1	87.5	93.3	90.3	–	–	–
MeCab+ER				**93.5**	**96.5**	**95.0**	**91.4**	**94.3**	**92.8**	55.9	25.8	35.3
Ours	✓			91.3	93.8	92.6	87.6	90.0	88.8	50.9	19.4	28.1
	✓	✓		91.0	93.7	92.3	88.1	90.8	89.4	42.4	28.0	33.8
	✓		✓	91.9	94.2	93.1	89.0	91.2	90.1	52.9	32.1	39.9
	✓	✓	✓	91.1	93.8	92.5	88.3	90.9	89.6	49.7	37.0	42.4
Ours +SP	✓	✓	✓	92.9	94.1	93.5	89.9	91.1	90.5	50.8	**37.8**	43.4
Ours +SP +NP	✓	✓	✓	92.9	94.1	93.5	89.9	91.1	90.5	**65.8**	36.6	**47.1**

Table 6.4. Precision (P), recall (R), and F$_1$ score of the baseline and proposed methods for word segmentation, POS tagging, and lexical normalization.

6.4.6. Evaluation Metrics

We used word-level precision, recall, and F$_1$ score to evaluate systems on each task. As shown in Table 2.3, a test word has labels corresponding to an index pair of the first and last character, i.e., span, a POS tag, and a standard form set. A predicted word w is counted as a TP when the span of w equals to that of a test word for the segmentation task and when the span and POS tag of w equal to those of a test word for the POS tagging task. For the normalization task, a predicted word w is counted as a TP when the span of w equals to that of a test word and the normalized form of w is included in the standard form set of the test word, whereas w is counted as a FP when either of the span or normalized form of w does not match with a test nonstandard word. A test word w is counted as a false negative when the span and normalized form of any predicted word do not match with w.

6.5. Results and Analysis

6.5.1. Main Results

Table 6.4 shows the performance of the proposed model (with the full features, unless otherwise specified) on the three tasks. Although the proposed model trained only on $\mathcal{A}_t$ achieved low normalization recall, the model with additional data $\mathcal{A}_d$ or $\mathcal{A}_r$ achieved a higher score, and the model with the three datasets achieved the highest score. These results are roughly consistent with the observation that

Training data	All		Nonstandard	
	Test OOV		Test OOV	
	#	%	#	%
$\emptyset$	12,600	100.0	767	100.0
$\mathcal{A}_t$ ($\mathcal{D}_t$)	1,055	8.4	445	58.0
$\mathcal{A}_t \cup \mathcal{A}_r$	734	5.8	335	43.7
$\mathcal{A}_t \cup \mathcal{A}_d$	764	6.1	343	44.7
$\mathcal{A}_t \cup \mathcal{A}_d \cup \mathcal{A}_r$	636	5.1	284	37.0

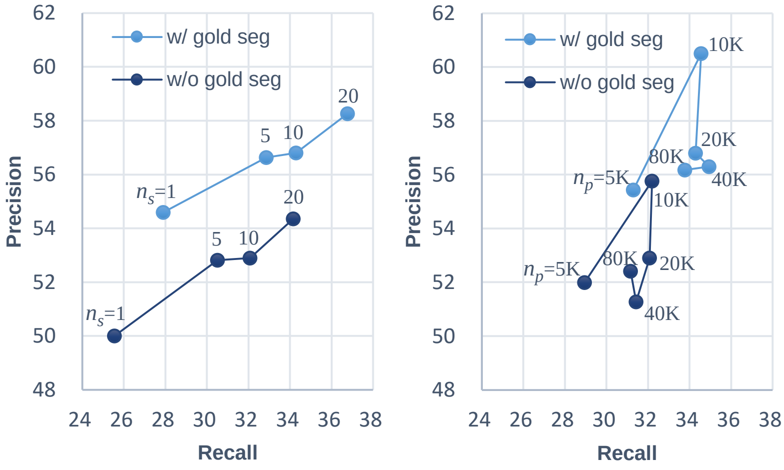
Table 6.5. The number and percentage of (all and nonstandard) test OOV tokens for each training dataset.

adding different types of pseudo-labeled data reduced the number of OOV tokens in the test data, as shown in Table 6.5. Our model with post-processing achieved further improvements; SP improved F_1 for each task by approximately 1 point and NP improved normalization precision by 15 points. The latter results indicate that avoiding the predictions of meaningless or unusual forms has the potential to improve our model.

We also evaluated the two existing methods for comparison: MeCab and MeCab+ER. Compared with MeCab+ER, our model achieved better normalization performance when trained on sufficient training data. Conversely, MeCab+ER achieved the best segmentation and POS tagging performance. The superiority of MeCab+ER over MeCab indicates the advantage of the explicit prediction of normalized word spans by the method on the two tasks, which contrasts with the independent prediction of word spans and normalized forms performed by our model.

6.5.2. Effect of Dataset Size

To investigate the effect of the amount of pseudo-labeled data, we generated dictionary-derived data $\mathcal{A}'_d$ with different settings of n_s and n_p, where n_s is the number of extracted sentences per variant pair and n_p is the number of variant pairs, as described in §6.4.1. We then evaluated the normalization performance



(i) Varying n_s ($n_p = 20\text{K}$) (ii) Varying n_p ($n_s = 10$)

Figure 6.2. Normalization performance of the proposed model trained on $\mathcal{A}_t \cup \mathcal{A}'_d$ with different size settings.

of the proposed model trained on $\mathcal{A}_t \cup \mathcal{A}'_d$.[72]

Figure 6.2 (i) shows the performance of the model with varying n_s and fixed $n_p = 20\text{K}$. A larger n_s led to both better precision and better recall, indicating that training with the same variant pairs but with more diverse context sentences contributed to more robust prediction. Figure 6.2 (ii) shows the performance of the model with fixed $n_s = 10$ and varying n_p. Increasing n_p from 5K to larger values contributed to better performance but increasing n_p above 10K did not improve recall, and degraded precision in most cases, probably because of the infrequent and ineffective variant pairs. Although the frequencies of the 5K-th and 10K-th nonstandard words in the constructed variant pairs were six and two, respectively, entries ranked below approximately 16K-th had a frequency of zero.[73]

These two results suggest that the gain discussed in §6.5.1 was caused by both the additional variant pairs and the additional contexts of existing variant pairs in the combined data of $\mathcal{A}_d$ and $\mathcal{A}_r$.

[72] We also evaluated the model's performance given gold segmentation to remove the influence of segmentation errors, but the model showed a similar tendency regardless of whether gold segmentation was given, as shown in Figure 6.2.

[73] Different entries with the same frequency were sorted in Japanese alphabetical order.

Feature	SEdit (Keep)		CConv (Keep)		Norm-neg	
	P	R	P	R	P	R
Char	97.4	99.5	98.2	99.0	97.2	98.7
Char+Lex	97.2	99.6	98.4	99.0	97.0	98.8
Char+Lex+Pron	97.4	99.6	98.5	99.2	97.3	99.0

Feature	SEdit (other)		CConv (other)		Norm	
	P	R	P	R	P	R
Char	77.5	39.5	56.7	41.5	48.6	36.9
Char+Lex	77.7	34.0	60.6	48.0	50.4	35.6
Char+Lex+Pron	80.5	39.8	67.9	52.2	54.6	40.3

Table 6.6. Precision (P) and recall (R) of the proposed models with character (Char), lexicon (Lex), and pronunciation (Pron) features for character-level tag prediction (SEdit/CConv) and word-level text editing (Norm-neg/Norm).

6.5.3. Detailed Results of Normalization

We semi-automatically annotated the test sentences with SEdit and CConv tags. Of 767 nonstandard tokens, six and three words required multi-character edit operations and replacement by Roman letters, respectively. Therefore, the upper bound of normalization recall was 99% by our tag set.

We then evaluated the proposed model with different features trained on the full $\mathcal{A}_t \cup \mathcal{A}_d \cup \mathcal{A}_r$ dataset, with respect to the character-level SEdit/CConv tag prediction accuracy for KEEP and other tags, and the word-level text editing accuracy of negative and positive normalization instances, which is calculated according to gold segmentation. A negative instance indicates a gold word annotated with no standard forms, and a prediction is regarded as correct when only KEEP tags are predicted for the word. Notably, in the test data, KEEP tags account for 95.8% and 96.8% of all SEdit and CConv tags, respectively, and negative instances account for 93.9% of all word tokens.

As shown in Table 6.6, for KEEP tags and negative normalization instances, the three models with different features achieved high recall (close to, or better than, 99%) and somewhat lower precision (around 97%–98%), indicating that undetected nonstandard words and over-normalized words accounted for the remaining 1% and 2%–3%, respectively. For other tags and positive normalization

Feature	IV	OOV
Char	54.3	11.8
Char+Lex	55.8	7.3
Char+Lex+Pron	62.1	11.8

Table 6.7. Recall of the proposed model with character (Char), lexicon (Lex), and pronunciation (Pron) features for in-vocabulary (IV) and out-of-vocabulary (OOV) normalization instances.

instances, both character-level and word-level performance was much lower because of the small number of training instances. However, the models with more additional features achieved better performance, indicating the effectiveness of the lexicon and pronunciation features. In particular, each additional feature substantially improved the performance of CConv tag prediction.

6.5.4. Performance for In-Vocabulary and Out-of-Vocabulary Normalization Instances

Letting a token be in-vocabulary (IV) if the token and its gold standard form[74] are included in the full training data $\mathcal{A}_d \cup \mathcal{A}_r \cup \mathcal{A}_t$ and be OOV otherwise, normalization instances in the test data consist of 385 IV and 382 OOV instances. Table 6.7 shows the recall of the proposed models trained on the full dataset with different features for both type of instances. Unsurprisingly, all model variations recognized IV instances much better than OOV instances. Although the model with full features achieved the highest recall of 62.1%, this is lower than the model's recall of 86.0% on the (pseudo) development data that only included IV normalization instances. The performance difference for IV instances and development instances is likely because of more distant context distribution of test instances from training instances.

[74] We regard a form converted from the original token by the annotated tags as the gold standard form.

Type	Gold	Det	ValTag	CorSeg	CorKC	Recall
Sound change variants	419	255	164	156	156	37.2
Character type variants	248	149	105	94	92	37.1
Alternative representations	132	78	55	51	51	38.6
Typographical errors	23	3	1	1	1	4.4
False positive	–	121	–	–	–	–

Table 6.8. Evaluation of the proposed model for each category defined in §5.2. "False positive" indicates the number of words detected incorrectly by the model.

6.5.5. Error Analysis

We conducted a step-by-step evaluation of the proposed model trained on the full dataset. Table 6.8 shows the number of gold normalization instances (Gold), predictions correctly detected (Det) out of Gold, predictions with valid SEdit/CConv tags (ValTag) out of Det, predictions correctly segmented (CorSeg) out of ValTag, and predictions matched with correct standard forms after text editing and KC (CorKC) out of CorSeg, for each category. For each of the top three categories (sound change variants, character type variants, alternative representations), two major errors were detection failures, which accounted for 39%–41% ($\frac{\text{Gold}-\text{Det}}{\text{Gold}}$), and tag prediction errors, which accounted for 17%–22% ($\frac{\text{Det}-\text{ValTag}}{\text{Gold}}$), whereas TPs accounted for 37%–39% ($\text{Recall}=\frac{\text{CorKC}}{\text{Gold}}$). Thus, our model achieved similar recognition performance for nonstandard words of the three categories except for typographical errors, which were rarely covered by the generated pseudo-labeled data, and recognized instances occurred with more diverse context in the pseudo-labeled data better according to the previous experiments described in §6.5.2 and §6.5.4. Notably, the proposed model achieved better recall than MeCab+ER (shown in Table 5.6) for all categories. As the most notable difference, our model correctly normalized 37% of character type variants, whereas MeCab+ER completely failed to normalize them.

Table 6.9 shows similar evaluation results for KC, where DetKC indicates the number of gold words assigned `TO_KANJI` tag(s) by the model. We found that most errors for the "required" instances were detection failures, and the KC model correctly converted 97% ($\frac{\text{CorKC}}{\text{CorSeg}}$) of the "required" and "optional" instances.

With respect to the precision degradation, the model over-normalized 121 neg-

Type	Gold	DetKC	ValTag	CorSeg	CorKC
Required	116	58	52	49	48
Optional	170	22	21	21	20
False positive	–	47	–	–	–

Table 6.9. Evaluation of the proposed model for kanji conversion (KC). “Required” indicates that any standard forms of a nonstandard word require KC. “Optional” indicates that some standard forms require KC but other standard forms can be generated without KC. “False positive” indicates the number of incorrect detection by the model.

ative instances (FPs) including 61 cases that were interjections or onomatopoeic words.[75] Examples that were over-normalized by the model trained on the full dataset are shown in Table 6.10. Both interjections and onomatopoeic words have characteristics similar to those of general nonstandard forms, such as insertion of Japanese special mora characters (a and e–h in Table 6.10), use of lowercased kana characters (d–e), and repetition of the same characters (d and h). These characteristics made it difficult to distinguish negative instances from words to be normalized. Another 29 cases were somewhat informal forms written in katakana or hiragana (i–n) and approximately 60% of the predicted normalized forms were acceptable, according to our assessment. This is because of the difficulty of annotating all words in a test sentence with all possible lexical variations. Incorrect normalization results included cases with peculiar spellings where some hiragana or katakana characters were converted to another type of kana (c and m–n).

6.6. Related Work

Text Editing. Text editing methods have also been applied to English lexical normalization. Chrupała [18] used character embeddings based on an RNNLM and trained CRFs to predict character-level edit operations. Min and Mott [78] proposed an LSTM-based model to perform word-level edit operations that aggregate character-level edit operations, e.g., a word-level tag “insert_h_replace_t” transforms “dese” into “these”.

[75] Interjections except idiomatic greetings and most onomatopoeic words were not annotated with any standard forms on the basis of the annotation criteria described in §5.4.1.

	Type	Original word	Edited word	KC result	Assess
(a)	Onomatopoeia	ガコンッ *gakon* (thud)	ガコン	–	✓
(b)	Onomatopoeia	ジンジン *jinjin* (tingling)	じんじん	–	✓
(c)	Onomatopoeia	ゴホゴホ *gohogoho* (coughing sound)	ごほごホ	–	?
(d)	Interjection	はぁぁ *haā* (sighing sound)	はああ	–	✓
(e)	Interjection	ぅん *un* (yeah)	うん	–	✓
(f)	Interjection	ひーひっひー *hīhihhī* (evil laugh sound)	ひいひっひい	–	✓
(g)	Interjection	あらっ *ara* (oh)	あらい *arai*	洗い (wash)	×
(h)	Interjection	おお〜〜 *ō* (wow)	おう	王 (king)	×
(i)	Informal	ケータイ *kētai* (mobile phone)	ケイタイ	–	✓
(j)	Informal	ダルい *darui* (dull/tired)	だるい	–	✓
(k)	Informal	キズ *kizu* (wound)	キズ	傷	✓
(l)	Informal	かっこ［Ｅ］ *kakko*[*ī*] (cool)	かっこ	カッコ	✓
(m)	Informal	ムレる *mureru* (stuffy)	むレる	–	?
(n)	Informal	ヤバイ *yabai* (crazy)	やバイ	–	?

Table 6.10. Examples of over-normalization by the proposed model. Words in "[]" indicate the surrounding context. "Edited word" shows the model's output after editing according to predicted tags, and "KC result" shows the result after performing kanji conversion (KC). "Assess" shows our assessments: "✓", "?", and "×" indicate that the final output is acceptable (the meaning is mostly preserved), questionable (the meaning is understandable but the spelling is peculiar), and obviously incorrect, respectively.

Recently, text editing models based on Transformer and BERT [73, 74, 121] have been proposed for monolingual sequence transduction tasks, such as grammatical error correction and text normalization for speech synthesis, because of their sample-efficient and fast inference characteristics compared to sequence-to-sequence models.

Data Synthesis. Data synthesis and augmentation methods have been explored for various NLP tasks, to increase the diversity of training examples [30] and for lexical normalization to address the deficiency of training data. Ikeda et al. [47] synthesized Japanese formal-informal sentence pairs by hand-crafted rules to convert standard forms to nonstandard forms. Zhang et al. [164] synthesized training data for Chinese informal word detection by random substitution of formal words in segmented sentences by informal words in a dictionary of formal-informal word pairs. To train statistical and neural MT models for Turkish text normalization, Çolakoğlu [19] generated a pseudo-parallel corpus where nonstandard words in original tweet text were aligned with plausible standard words using their weighted edit distance algorithm. Dekker and van der Goot [22] explored data synthesis methods for English lexical normalization using the clean-to-noisy policy (based on manually-designed rules) and the noisy-to-clean policy (based on predicted standard forms).

6.7. Conclusion

In this chapter, we proposed a text editing model and methods of pseudo-labeled data generation for the joint segmentation, POS tagging, and normalization task. The experiments demonstrated that the proposed model was successfully trained on generated pseudo-labeled data and achieved better normalization performance than the existing method, which was evaluated in Chapter 5. However, more exhaustive detection and accurate normalization of nonstandard words have the potential to improve the model.

7. Conclusions

7.1. Summary

This thesis presented our work on Japanese and Chinese word segmentation and Japanese lexical normalization.

First, we proposed a neural character-based segmenter that integrated word-level information via an attention mechanism. The experimental results showed that the proposed model with attention-weighted word vectors improved the performance in in-domain (as well as in cross-domain) Japanese and Chinese word segmentation.

Second, we proposed a method using unlabeled data and lexicons for cross-domain word segmentation, incorporating target-domain knowledge into a neural segmenter as an auxiliary prediction task to identify word occurrences in unlabeled sentences. The experiments showed that the auxiliary task improved the performance for the Japanese and Chinese target domains, while achieving similar or better performance for source and other domains.

Third, we constructed a publicly available Japanese UGT dataset annotated with morphological, normalization, and word category information. The experiments on our corpus showed the limited performance of existing systems for non-general and nonstandard words.

Finally, we proposed a neural text editing model and methods of pseudo-labeled data generation for the joint task of Japanese word segmentation, POS tagging, and lexical normalization (SPN). The experiments showed that the proposed model trained on generated data achieved better normalization performance than an existing normalization method, particularly when multiple types of pseudo-labeled data were combined.

Through our work, we demonstrated that our proposed segmentation and normalization methods achieved performance better than or competitive to existing state-of-the-art methods, and our evaluation dataset can be used as a benchmark to compare and analyze different systems.

7.2. Future Directions

There are several directions for future work on word segmentation and lexical normalization.

Model Size and Inference Speed. Although fast and lightweight segmentation is desirable as a preprocessing step for downstream NLP tasks, neural segmentation methods tend to have larger model size and slower inference speed than traditional non-neural statistical methods. Knowledge distillation from a large model with high accuracy to a small model is a prospective approach.

For example, Huang et al. [44] used knowledge distillation through pseudo labels from a teacher model, which is a fine-tuned pre-trained language model (PLM), into fast and lightweight student models, which includes neural and non-neural models, and achieved competitive performance in comparison to state-of-the-art Chinese word segmenters.

Performance Improvement on UGT Processing. Although our work on the Japanese SPN task demonstrated the performance improvements of the proposed method by adding features and increasing training data, the absolute performance is still low, i.e., at most F_1 of 93% for segmentation and 42% for normalization. For more accurate segmentation and more accurate and exhausted normalization, possible enhancements include explicit modeling of the span prediction of nonstandard words and the incorporation of knowledge from large PLMs.

For example, Samuel and Straka [110] proposed a system based on ByT5 [151], which is a multilingual byte-level PLM, to conduct span-level masking and prediction. Their system achieved the best performance on 11 languages, including English, German, and Italian, at the W-NUT 2021 multilingual lexical normalization shared task.[76]

[76] http://noisy-text.github.io/2021/multi-lexnorm.html

Evaluation on Broader UGT Domains and Phenomena. We evaluated the existing and proposed system performance in two UGT domains: blog and Q&A forum. Constructing evaluation data in various UGT domains is beneficial for evaluating the performance of segmentation/normalization methods for non-canonical expressions that frequently occur in other (more biased) UGT domains, such as social media posts and movie site comments. Such expressions may include proper names, neologisms, and dialect words, which our work did not focus on because of the low frequencies in the constructed corpus. In addition, corpus construction and system development for segmentation and normalization of UGT in other unsegmented languages, such as Chinese, are also possible future directions.

References

[1] Masayuki Asahara, Hiroshi Kanayama, Yusuke Miyao, Takaaki Tanaka, Mai Omura, Yugo Murawaki, and Yuji Matsumoto. Japanese universal dependencies corpora [in Japanese]. *Journal of Natural Language Processing*, 26(1):3–36, 2019.

[2] Masayuki Asahara and Yuji Matsumoto. Extended models and tools for high-performance part-of-speech. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*, 2000.

[3] Masayuki Asahara and Yuji Matsumoto. Japanese unknown word identification by character-based chunking. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 459–465, Geneva, Switzerland, 2004. COLING.

[4] Kurohashi Laboratory at Kyoto University. Japanese morphological analysis system JUMAN 7.0 users manual, 2012.

[5] AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. A phrase-based statistical model for SMS text normalization. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 33–40, Sydney, Australia, 2006. Association for Computational Linguistics.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, Conference Track Proceedings*, San Diego, CA, USA, 2015.

[7] Timothy Baldwin, Marie Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition.

In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135, Beijing, China, 2015. Association for Computational Linguistics.

[8] Deng Cai and Hai Zhao. Neural word segmentation learning for Chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–420, Berlin, Germany, 2016. Association for Computational Linguistics.

[9] Deng Cai, Hai Zhao, Zhisong Zhang, Yuan Xin, Yongjian Wu, and Feiyue Huang. Fast and accurate neural word segmentation for Chinese. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 608–615, Vancouver, Canada, 2017. Association for Computational Linguistics.

[10] Zerui Cai, Taolin Zhang, Chengyu Wang, and Xiaofeng He. EMBERT: A pre-trained language model for Chinese medical text mining. In *Asia-Pacific Web and Web-Age Information Management Joint International Conference on Web and Big Data*, pages 242–257, Guangzhou, China, 2021. Springer, Cham.

[11] Keh-Jiann Chen and Wei-Yun Ma. Unknown word extraction for Chinese documents. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[12] Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 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*, pages 3960–3966, Melbourne, Australia, 2017. AAAI Press.

[13] Xinchi Chen, Xipeng Qiu, Chenxi Zhu, and Xuanjing Huang. 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 (Volume 1: Long Papers)*, pages 1744–1753, Beijing, China, 2015. Association for Computational Linguistics.

[14] Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. Long short-term memory neural networks for Chinese word segmentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1197–1206, Lisbon, Portugal, 2015. Association for Computational Linguistics.

[15] Fei Cheng, Shuntaro Yada, Ribeka Tanaka, Eiji Aramaki, and Sadao Kurohashi. JaMIE: A pipeline Japanese medical information extraction system. *Computing Research Repository*, arXiv:2111.04261, 2021.

[16] Jason P.C. Chiu and Eric Nichols. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.

[17] Monojit Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, and Anupam Basu. Investigation and modeling of the structure of texting language. *International Journal of Document Analysis and Recognition*, 10(3):157–174, 2007.

[18] Grzegorz Chrupała. Normalizing tweets with edit scripts and recurrent neural embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 680–686, Baltimore, Maryland, USA, 2014. Association for Computational Linguistics.

[19] Talha Çolakoğlu, Umut Sulubacak, and Ahmet Cüneyd Tantuğ. Normalizing non-canonical Turkish texts using machine translation approaches. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 267–272, Florence, Italy, 2019. Association for Computational Linguistics.

[20] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167, Helsinki Finland, 2008. Association for Computing Machinery.

[21] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(76):2493–2537, 2011.

[22] Kelly Dekker and Rob van der Goot. Synthetic data for English lexical normalization: How close can we get to manually annotated data? In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6300–6309, Marseille, France, 2020. European Language Resources Association.

[23] Yasuharu Den. A multi-purpose electronic dictionary for morphological analyzers [in Japanese]. *Journal of the Japanese Society for Artificial Inteligence*, 34(5):640–646, 2009.

[24] Yasuharu Den, Junpei Nakamura, Toshinobu Ogiso, and Hideki Ogura. A proper approach to Japanese morphological analysis: Dictionary, model, and evaluation. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco, 2008. European Language Resources Association.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, USA, 2019. Association for Computational Linguistics.

[26] Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. ZEN: pre-training chinese text encoder enhanced by n-gram representations. *Computing Research Repository*, 1911.00720, 2019.

[27] Ning Ding, Dingkun Long, Guangwei Xu, Muhua Zhu, Pengjun Xie, Xiaobin Wang, and Haitao Zheng. Coupling distant annotation and adversarial training for cross-domain Chinese word segmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6662–6671, Online, 2020. Association for Computational Linguistics.

[28] Thomas Emerson. The 2nd international Chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, 2005.

[29] Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. Accessor variety criteria for Chinese word extraction. *Computational Linguistics*, 30(1):75–93, 2004.

[30] Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online, 2021. Association for Computational Linguistics.

[31] Leilei Gan and Yue Zhang. Investigating self-attention network for Chinese word segmentation. *Computing Research Repository*, arXiv:1907.11512, 2019.

[32] Laurence Gillick and Stephen J Cox. Some statistical issues in the comparison of speech recognition algorithms. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 532–535. IEEE, 1989.

[33] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional LSTM. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278, 2013.

[34] Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuanjing Huang. A lexicon-based graph neural network for Chinese NER. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1040–1050, Hong Kong, 2019. Association for Computational Linguistics.

[35] Bo Han and Timothy Baldwin. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*,

pages 368–378, Portland, Oregon, USA, 2011. Association for Computational Linguistics.

[36] Masatsugu Hangyo, Daisuke Kawahara, and Sadao Kurohashi. Building a diverse document leads corpus annotated with semantic relations. In *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*, pages 535–544, Bali, Indonesia, 2012. Faculty of Computer Science, Universitas Indonesia.

[37] Chikara Hashimoto, Sadao Kurohashi, Daisuke Kawahara, Keiji Shinzato, and Masaaki Nagata. Construction of a blog corpus with syntactic, anaphoric, and sentiment annotations. *Journal of Natural Language Processing*, 18(2):175–201, 2011.

[38] Shohei Higashiyama, Masao Utiyama, Yuji Matsumoto, Taro Watanabe, and Eiichiro Sumita. Auxiliary lexicon word prediction for cross-domain word segmentation. *Journal of Natural Language Processing*, 27(3):573–598, 2020.

[39] Shohei Higashiyama, Masao Utiyama, Eiichiro Sumita, Masao Ideuchi, Yoshiaki Oida, Yohei Sakamoto, and Isaac Okada. 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, Volume 1 (Long and Short Papers)*, pages 2699–2709, Minneapolis, Minnesota, USA, 2019. Association for Computational Linguistics.

[40] Shohei Higashiyama, Masao Utiyama, Eiichiro Sumita, Masao Ideuchi, Yoshiaki Oida, Yohei Sakamoto, Isaac Okada, and Yuji Matsumoto. Character-to-word attention for word segmentation. *Journal of Natural Language Processing*, 27(3):499–530, 2020.

[41] Shohei Higashiyama, Masao Utiyama, Taro Watanabe, and Eiichiro Sumita. A text editing approach to joint Japanese word segmentation, POS tagging, and lexical normalization. In *Proceedings of the Seventh Workshop on Noisy User-generated Text*, pages 67–80, Online, 2021. Association for Computational Linguistics.

[42] Shohei Higashiyama, Masao Utiyama, Taro Watanabe, and Eiichiro Sumita. User-generated text corpus for evaluating Japanese morphological analysis and lexical normalization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5532–5541, Online, 2021. Association for Computational Linguistics.

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

[44] Kaiyu Huang, Junpeng Liu, Degen Huang, Deyi Xiong, Zhuang Liu, and Jinsong Su. Enhancing Chinese word segmentation via pseudo labels for practicability. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4369–4381, Online, 2021. Association for Computational Linguistics.

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

[46] Kazushi Ikeda, Tadashi Yanagihara, Kazunori Matsumoto, and Yasuhiro Takishima. Automatic rule generation approach for morphological analysis of peculiar expressions on blog documents [in Japanese]. *IPSJ Transactions on Databases*, 3(3):68–77, 2010.

[47] Taishi Ikeda, Hiroyuki Shindo, and Yuji Matsumoto. Japanese text normalization with encoder-decoder model. In *Proceedings of the 2nd Workshop on Noisy User-generated Text*, pages 129–137, Osaka, Japan, 2016. The COLING 2016 Organizing Committee.

[48] Wenbin Jiang, Meng Sun, Yajuan Lü, Yating Yang, and Qun Liu. Discriminative learning with natural annotations: Word segmentation as a case study. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 761–769, Sofia, Bulgaria, 2013. Association for Computational Linguistics.

[49] Ning Jin. NCSU-SAS-ning: Candidate generation and feature engineering for supervised lexical normalization. In *Proceedings of the Workshop on*

Noisy User-generated Text, pages 87–92, Beijing, China, 2015. Association for Computational Linguistics.

[50] Nobuhiro Kaji and Masaru Kitsuregawa. Accurate morphological analysis by jointly performing lexical normalization. *JSAI Technical Report, SIG-FPAI*, 93:99–104, 2014.

[51] Nobuhiro Kaji and Masaru Kitsuregawa. Accurate word segmentation and pos tagging for Japanese microblogs: Corpus annotation and joint modeling with lexical normalization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 99–109, Doha, Qatar, 2014. Association for Computational Linguistics.

[52] Nobuhiro Kaji, Shinsuke Mori, Fumihiko Takahashi, Tetsuro Sasada, Itsumi Saito, Keigo Hattori, Yugo Murawaki, and Kei Utsumi. Kētaiso kaiseki no error bunseki (Error analysis of morphological analysis) [in Japanese]. In *Proceedings of the 21th Annual Meeting of the Association for Natural Language Processing Workshop "Error Analysis on Natural Language Processing"*, Kyoto, Japan, 2015.

[53] Yoshiaki Kitagawa and Mamoru Komachi. Long short-term memory for Japanese word segmentation. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, pages 279–288, Hong Kong, 2018. Association for Computational Linguistics.

[54] Aomi Koyama, Tomoshige Kiyuna, Kenji Kobayashi, Mio Arai, and Mamoru Komachi. Construction of an evaluation corpus for grammatical error correction for learners of Japanese as a second language. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 204–211, Marseille, France, 2020. European Language Resources Association.

[55] Shunsuke Kozawa, Kiyotaka Uchimoto, and Yasuharu Den. Adaptation of long-unit-word analysis system to different part-of-speech tagset. *Journal of Natural Language Processing*, 21(2):379–401, 2014.

[56] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 66–75, Melbourne, Australia, 2018. Association for Computational Linguistics.

[57] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237, Barcelona, Spain, 2004. Association for Computational Linguistics.

[58] Sadao Kurohashi and Makoto Nagao. Building a Japanese parsed corpus. In *Treebanks*, pages 249–260. Springer, 2003.

[59] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, 2001.

[60] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[61] Zhifei Li and David Yarowsky. Mining and modeling relations between formal and informal Chinese phrases from web corpora. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1031–1040, Honolulu, Hawaii, 2008. Association for Computational Linguistics.

[62] Ming-Yu Lin, Tung-Hui Chiang, and Keh-Yih Su. A preliminary study on unknown word problem in Chinese word segmentation. In *Proceedings of Rocling VI Computational Linguistics Conference VI*, pages 119–141, Nantou, Taiwan, September 1993. The Association for Computational Linguistics and Chinese Language Processing.

[63] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 2124–2133, Berlin, Germany, 2016. Association for Computational Linguistics.

[64] Fei Liu, Fuliang Weng, Bingqing Wang, and Yang Liu. Insertion, deletion, or substitution? Normalizing text messages without pre-categorization nor supervision. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 71–76, Portland, Oregon, USA, 2011. Association for Computational Linguistics.

[65] Junxin Liu, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. Neural chinese word segmentation with lexicon and unlabeled data via posterior regularization. In *The World Wide Web Conference*, pages 3013–3019, San Francisco, California, USA, 2019.

[66] Yang Liu and Yue Zhang. Unsupervised domain adaptation for joint segmentation and POS-tagging. In *Proceedings of COLING 2012: Posters*, pages 745–754, Mumbai, India, 2012. The COLING 2012 Organizing Committee.

[67] Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, and Ting Liu. Exploring segment representations for neural segmentation models. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 2880–2886, New York, New York, USA, 2016. AAAI Press / International Joint Conferences on Artificial Intelligence.

[68] Yijia Liu, Yue Zhang, Wanxiang Che, Ting Liu, and Fan Wu. Domain adaptation for crf-based Chinese word segmentation using free annotations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 864–874, Doha, Qatar, 2014. Association for Computational Linguistics.

[69] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, 2015. Association for Computational Linguistics.

[70] Ji Ma, Kuzman Ganchev, and David Weiss. State-of-the-art Chinese word segmentation with Bi-LSTMs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4902–4908, Brussels, Belgium, 2018. Association for Computational Linguistics.

[71] Kikuo Maekawa. Corpus of spontaneous Japanese: Its design and evaluation. In *Proceedings of the ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, pages 7–12, 2003.

[72] Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. Balanced corpus of contemporary written Japanese. *Language Resources and Evaluation*, 48(2):345–371, 2014.

[73] Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. FELIX: Flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255, Online, 2020. Association for Computational Linguistics.

[74] Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5054–5065, Hong Kong, 2019. Association for Computational Linguistics.

[75] Mairgup Mansur, Wenzhe Pei, and Baobao Chang. Feature-based neural language model and Chinese word segmentation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1271–1277, Nagoya, Japan, 2013. Asian Federation of Natural Language Processing.

[76] Takasi Masuoka and Yukinori Takubo. *Kiso Nihongo Bumpō (Essential Japanese Grammar) [in Japanese]*. Kuroshio Shuppan, 1989.

[77] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, Workshop Track Proceedings*, Scottsdale, Arizona, USA, 2013.

[78] Wookhee Min and Bradford Mott. NCSU_SAS_WOOKHEE: A deep contextual long-short term memory model for text normalization. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 111–119, Beijing, China, 2015. Association for Computational Linguistics.

[79] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 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*, pages 1003–1011, Suntec, Singapore, 2009. Association for Computational Linguistics.

[80] Shotaro Misawa, Motoki Taniguchi, Yasuhide Miura, and Tomoko Ohkuma. Character-based bidirectional LSTM-CRF with words and characters for Japanese named entity recognition. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 97–102, Copenhagen, Denmark, 2017. Association for Computational Linguistics.

[81] Chiaki Miyazaki and Satoshi Sato. Classification of phonological changes reflected in text: Toward a characterization of written utterances [in Japanese]. *Journal of Natural Language Processing*, 26(2):407–440, 2019.

[82] Shinsuke Mori and Makoto Nagao. Word extraction from corpora and its part-of-speech estimation using distributional analysis. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*, 1996.

[83] Shinsuke Mori, Hideki Ogura, and Tetsuro Sasada. A Japanese word dependency corpus. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 753–758, Reykjavik, Iceland, 2014. European Language Resources Association.

[84] Makoto Morishita, Jun Suzuki, and Masaaki Nagata. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3603–3609, Marseille, France, 2020. European Language Resources Association.

[85] Hajime Morita, Daisuke Kawahara, and Sadao Kurohashi. Morphological analysis for unsegmented languages using recurrent neural network language model. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2297, Lisbon, Portugal, 2015. Association for Computational Linguistics.

[86] Takuro Moriyama and Katsumi Shibuya. *Meikai Nihongogaku Jiten (The Sanseido Dictionary of Japanese Linguistics) [in Japanese]*. Sanseido, 2020.

[87] Benjamin Muller, Benoit Sagot, and Djamé Seddah. Enhancing BERT for lexical normalization. In *Proceedings of the 5th Workshop on Noisy User-generated Text*, pages 297–306, Hong Kong, China, November 2019. Association for Computational Linguistics.

[88] Yugo Murawaki and Sadao Kurohashi. Online acquisition of Japanese unknown morphemes using morphological constraints. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 429–437, Honolulu, Hawaii, USA, 2008. Association for Computational Linguistics.

[89] Yugo Murawaki and Sadao Kurohashi. Online Japanese unknown morpheme detection using orthographic variation. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valletta, Malta, 2010. European Language Resources Association.

[90] Tetsuji Nakagawa. Chinese and Japanese word segmentation using word-level and character-level information. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 466–472, Geneva, Switzerland, 2004. COLING.

[91] Yasunari Nakamoto, Kazuya Mera, and Aizawa teruaki. Using sequence

alignment to improve the morphological analysis [in Japanese]. *IPSJ SIG Technical Report*, 2000(11):87–93, 2000.

[92] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, Portland, Oregon, USA, 2011. Association for Computational Linguistics.

[93] Hideki Ogura, Hanae Koiso, Yumi Fujiike, Sayaka Miyauchi, Hikari Konishi, and Yutaka Hara. Gendai nihongo kakikotoba kinkō corpus kētairon kitēshū dai 4 ban ge (Regulations of morphological information for balanced corpus of contemporary written Japanese 4th edition volume 2) [in Japanese]. *NINJAL Internal Reports*, 2011.

[94] Hideki Ogura, Hanae Koiso, Yumi Fujiike, Sayaka Miyauchi, Hikari Konishi, and Yutaka Hara. Gendai nihongo kakikotoba kinkō corpus kētairon kitēshū dai 4 ban jō (Regulations of morphological information for balanced corpus of contemporary written Japanese 4th edition volume 1) [in Japanese]. *NINJAL Internal Reports*, 2011.

[95] Takuya Okimori, Hajime Kimura, Norimasa Suzuki, and Mitsuhiro Yoshida. *Nihongo Library Go-to Goi (Japanese Language Library Word and Vocabulary) [in Japanese].* Asakura Publishing, 2012.

[96] Ayaha Osaki, Yoshiaki Kitagawa, and Mamoru Komachi. Nihongo Twitter bunsho wo taishō toshita kēretsu labeling niyoru hyōki sēkika (Text normalization by sequence labeling for Japanese Twitter documents) [in Japanese]. *IPSJ SIG Technical Report*, 2017-NL-231(12):1–6, 2017.

[97] Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. Neural modeling of multi-predicate interactions for Japanese predicate argument structure analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1591–1600, Vancouver, Canada, 2017. Association for Computational Linguistics.

[98] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas, USA, 2016. Association for Computational Linguistics.

[99] Wenzhe Pei, Tao Ge, and Baobao Chang. Max-margin tensor neural network for Chinese word segmentation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–303, Baltimore, Maryland, USA, 2014. Association for Computational Linguistics.

[100] Fuchun Peng, Fangfang Feng, and Andrew McCallum. Chinese segmentation and new word detection using conditional random fields. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 562–568, Geneva, Switzerland, 2004. COLING.

[101] Tao Qian, Yue Zhang, Meishan Zhang, Yafeng Ren, and Donghong Ji. A transition-based model for joint segmentation, POS-tagging and normalization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1837–1846, Lisbon, Portugal, 2015. Association for Computational Linguistics.

[102] Likun Qiu, Linlin Shi, and Houfeng Wang. Construction of multi-domain Chinese dependency treebanks and a study on factors influencing the statistical parsing. *Journal of Chinese Information Processing*, 29(5), 2015.

[103] Likun Qiu and Yue Zhang. Word segmentation for Chinese novels. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2440–2446, Austin, Texas, USA, 2015. AAAI Press.

[104] Vivek Kumar Rangarajan Sridhar. Unsupervised text normalization using distributed representations of words and phrases. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 8–16, Denver, Colorado, USA, 2015. Association for Computational Linguistics.

[105] Radim Řehůřek and Petr Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, 2010.

[106] Marek Rei, Gamal K.O. Crichton, and Sampo Pyysalo. Attending to characters in neural sequence labeling models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 309–318, Osaka, Japan, 2016. The COLING 2016 Organizing Committee.

[107] Itsumi Saito, Kyosuke Nishida, Kugatsu Sadamitsu, Kuniko Saito, and Junji Tomita. Automatically extracting variant-normalization pairs for Japanese text normalization. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 937–946, Taipei, Taiwan, 2017. Asian Federation of Natural Language Processing.

[108] Itsumi Saito, Kugatsu Sadamitsu, Hisako Asano, and Yoshihiro Matsuo. Morphological analysis for Japanese noisy text based on character-level and word-level normalization. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1773–1782, Dublin, Ireland, 2014. Dublin City University and Association for Computational Linguistics.

[109] Hiroki Sakaji, Ryota Kuramoto, Hiroyasu Matsushima, Kiyoshi Izumi, Takashi Shimada, and Keita Sunakawa. Financial text data analytics framework for business confidence indices and inter-industry relations. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*, pages 40–46, Macao, China, 2019.

[110] David Samuel and Milan Straka. ÚFAL at MultiLexNorm 2021: Improving multilingual lexical normalization by fine-tuning ByT5. In *Proceedings of the Seventh Workshop on Noisy User-generated Text*, pages 483–492, Online, 2021. Association for Computational Linguistics.

[111] Ryohei Sasano and Sadao Kurohashi. Japanese named entity recognition using structural natural language processing. In *Proceedings of the 3rd*

International Joint Conference on Natural Language Processing: Volume-II, 2008.

[112] Ryohei Sasano, Sadao Kurohashi, and Manabu Okumura. A simple approach to unknown word processing in Japanese morphological analysis. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pages 162–170, Nagoya, Japan, 2013. Asian Federation of Natural Language Processing.

[113] Ryoichi Sato. *Todōfuken betsu zenkoku hōgen jiten (Dialect Dictionary of Japanese Prefectures) [in Japanese]*. Sanseido, 2009.

[114] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal*, pages 5149–5152, Kyoto, Japan, 2012. IEEE.

[115] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, 2016. Association for Computational Linguistics.

[116] Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. Character-based joint segmentation and POS tagging for Chinese using bidirectional RNN-CRF. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 173–183, Taipei, Taiwan, 2017. Asian Federation of Natural Language Processing.

[117] Mo Shen, Daisuke Kawahara, and Sadao Kurohashi. Chinese word segmentation and unknown word extraction by mining maximized substring. *Journal of Natural Language Processing*, 23(3):235–266, 2016.

[118] Tomohide Shibata and Sadao Kurohashi. Entity-centric joint modeling of Japanese coreference resolution and predicate argument structure analysis.

In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 579–589, Melbourne, Australia, 2018. Association for Computational Linguistics.

[119] Cagil Sönmez and Arzucan Özgür. A graph-based approach for contextual text normalization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 313–324, Doha, Qatar, 2014. Association for Computational Linguistics.

[120] Richard Sproat and Thomas Emerson. The first international Chinese word segmentation bakeoff. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 133–143, Sapporo, Japan, 2003. Association for Computational Linguistics.

[121] Felix Stahlberg and Shankar Kumar. Seq2Edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 5147–5159, Online, 2020. Association for Computational Linguistics.

[122] Andreas Stolcke. Srilm—an extensible language modeling toolkit. In *7th International Conference on Spoken Language Processing*, 2002.

[123] Katsuhito Sudoh, Masaaki Nagata, Shinsuke Mori, and Tatsuya Kawahara. 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: MT Researchers Track*, pages 234–248, Vancouver, Canada, 2014. Association for Machine Translation in the Americas.

[124] Hiroaki Sugiyama, Masahiro Mizukami, Tsunehiro Arimoto, Hiromi Narimatsu, Yuya Chiba, Hideharu Nakajima, and Toyomi Meguro. Empirical analysis of training strategies of transformer-based Japanese chit-chat systems. *Computing Research Repository*, arXiv:2109.05217, 2021.

[125] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Proceedings of the 28th International Conference on Neural*

Information Processing Systems - Volume 2, pages 2440–2448, Montreal, Canada, 2015. MIT Press.

[126] Maosong Sun and Benjamin K. T'sou. Ambiguity resolution in Chinese word segmentation. In *Proceedings of the 10th Pacific Asia Conference on Language, Information and Computation*, pages 121–126, Hong Kong, 1995. City University of Hong Kong.

[127] Weiwei Sun and Jia Xu. Enhancing Chinese word segmentation using unlabeled data. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 970–979, Edinburgh, Scotland, UK, 2011. Association for Computational Linguistics.

[128] Zhiqing Sun, Gehui Shen, and Zhihong Deng. A gap-based framework for Chinese word segmentation via very deep convolutional networks. *Computing Research Repository*, arXiv:1712.09509, 2017.

[129] Kazuma Takaoka, Sorami Hisamoto, Noriko Kawahara, Miho Sakamoto, Yoshitaka Uchida, and Yuji Matsumoto. Sudachi: a Japanese tokenizer for business. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, Miyazaki, Japan, 2018. European Language Resources Association.

[130] Makoto Takenaka, Toshihiko Yanase, Atsuko Koizumi, and Yo Ehara. Tekitaiteki sēsē wo riyō shita kōsē yōhi no shikibetsu (sentence classification for proofreading using adversarial generation) [in Japanese]. *IPSJ SIG Technical Report*, 2018-NL-235(5):1–5, 2018.

[131] Koichi Takeuchi and Yuji Matsumoto. HMM parameter learning for Japanese morphological analyzer. In *Proceedings of the 10th Pacific Asia Conference on Language, Information and Computation*, pages 163–172, Hong Kong, 1995. City University of Hong Kong.

[132] Yuanhe Tian, Yan Song, Fei Xia, Tong Zhang, and Yonggang Wang. Improving Chinese word segmentation with wordhood memory networks. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8274–8285, Online, 2020. Association for Computational Linguistics.

[133] Arseny Tolmachev, Daisuke Kawahara, and Sadao Kurohashi. 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)*, pages 2744–2755, June 2019.

[134] Arseny Tolmachev, Daisuke Kawahara, and Sadao Kurohashi. Design and structure of the Juman++ morphological analyzer toolkit. *Journal of Natural Language Processing*, 27(1):89–132, 2020.

[135] Kiyotaka Uchimoto. Unknown word processing [in Japanese]. In *Digital Encyclopedia of Natural Language Processing*, pages 150–151. Kyoritsu Shuppan, 2010.

[136] Kiyotaka Uchimoto and Hitoshi Isahara. Morphological annotation of a large spontaneous speech corpus in Japanese. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1731–1737, Hyderabad India, 2007.

[137] Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara. 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*, pages 91–99, 2001.

[138] Rob van der Goot. MoNoise: A multi-lingual and easy-to-use lexical normalization tool. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 201–206, Florence, Italy, 2019. Association for Computational Linguistics.

[139] Rob van der Goot, Rik van Noord, and Gertjan van Noord. A taxonomy for in-depth evaluation of normalization for user generated content. In *Proceedings of the Eleventh International Conference on Language Resources*

and Evaluation, Miyazaki, Japan, 2018. European Language Resources Association.

[140] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 2017.

[141] Aobo Wang, Min-Yen Kan, Daniel Andrade, Takashi Onishi, and Kai Ishikawa. Chinese informal word normalization: an experimental study. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 127–135, Nagoya, Japan, 2013. Asian Federation of Natural Language Processing.

[142] Chunqi Wang and Bo Xu. Convolutional neural network with word embeddings for Chinese word segmentation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 163–172, Taipei, Taiwan, 2017. Asian Federation of Natural Language Processing.

[143] Pidong Wang and Hwee Tou Ng. A beam-search decoder for normalization of social media text with application to machine translation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 471–481, Atlanta, Georgia, USA, 2013. Association for Computational Linguistics.

[144] Xiaobin Wang, Deng Cai, Linlin Li, Guangwei Xu, Hai Zhao, and Luo Si. Unsupervised learning helps supervised neural word segmentation. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 7200–7207. AAAI Press, 2019.

[145] Andi Wu. Customizable segmentation of morphologically derived words in Chinese. *International Journal of Computational Linguistics & Chinese Language Processing: Special Issue on Word Formation and Chinese Language Processing*, 8(1):1–28, 2003.

[146] Guohua Wu, Dezhu He, Keli Zhong, Xue Zhou, and Caixia Yuan. Leveraging rich linguistic features for cross-domain Chinese segmentation. In *Proceedings of the 3rd CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 101–107, Wuhan, China, 2014. Association for Computational Linguistics.

[147] Fei Xia. The segmentation guidelines for the penn chinese treebank (3.0). *University of Pennsylvania Institute for Research in Cognitive Science Technical Report*, (IRCS-00-06), 2000.

[148] Fei Xia, Martha Palmer, Nianwen Xue, Mary Ellen Okurowski, John Kovarik, Fu-Dong Chiou, Shizhe Huang, Tony Kroch, and Mitch Marcus. Developing guidelines and ensuring consistency for Chinese text annotation. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, Athens, Greece, 2000. European Language Resources Association.

[149] Fengshun Xiao, Jiangtong Li, Hai Zhao, Rui Wang, and Kehai Chen. Lattice-based transformer encoder for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3090–3097, Florence, Italy, 2019. Association for Computational Linguistics.

[150] Jingjing Xu and Xu Sun. Dependency-based gated recursive neural network for Chinese word segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–572, Berlin, Germany, 2016. Association for Computational Linguistics.

[151] Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Computing Research Repository*, 2105.13626, 2021.

[152] 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):207–238, 2005.

[153] Nianwen Xue. Chinese word segmentation as character tagging. *International Journal of Computational Linguistics & Chinese Language Processing: Special Issue on Word Formation and Chinese Language Processing*, 8(1):29–48, 2003.

[154] Nakami Yamaguchi. *Giongo gitaigo jiten (Phonomime and Phenomime Dictionary) [in Japanese]*. Kodansha, 2002.

[155] Takahiro Yamakoshi, Takahiro Komamizu, Yasuhiro Ogawa, and Katsuhiko Toyama. Japanese mistakable legal term correction using infrequency-aware BERT classifier. In *2019 IEEE International Conference on Big Data*, pages 4342–4351, Los Angeles, CA, USA, 2019. IEEE.

[156] Jie Yang, Yue Zhang, and Fei Dong. Neural word segmentation with rich pretraining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 839–849, Vancouver, Canada, 2017. Association for Computational Linguistics.

[157] Jie Yang, Yue Zhang, and Shuailong Liang. 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, Volume 1 (Long and Short Papers)*, pages 2720–2725, Minneapolis, Minnesota, USA, 2019. Association for Computational Linguistics.

[158] Yaqin Yang and Nianwen Xue. Chinese comma disambiguation for discourse analysis. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 786–794, Jeju Island, Korea, 2012. Association for Computational Linguistics.

[159] Yi Yang and Jacob Eisenstein. A log-linear model for unsupervised text normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 61–72, Seattle, Washington, USA, 2013. Association for Computational Linguistics.

[160] Yuxiao Ye, Weikang Li, Yue Zhang, Likun Qiu, and Jian Sun. Improving cross-domain chinese word segmentation with word embeddings. In *Pro-*

ceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2726–2735, Minneapolis, Minnesota, USA, 2019. Association for Computational Linguistics.

[161] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *Computing Research Repository*, 1409.2329, 2014.

[162] Hua-Ping Zhang, Qun Liu, Xue-Qi Cheng, Hao Zhang, and Hong-Kui Yu. Chinese lexical analysis using hierarchical hidden Markov model. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 63–70, Sapporo, Japan, 2003. Association for Computational Linguistics.

[163] Longkai Zhang, Li Li, Zhengyan He, Houfeng Wang, and Ni Sun. 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)*, pages 177–182, 2013.

[164] Meishan Zhang, Guohong Fu, and Nan Yu. Segmenting Chinese microtext: Joint informal-word detection and segmentation with neural networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4228–4234, Melbourne, Australia, 2017. AAAI Press.

[165] Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 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*, pages 588–597, Gothenburg, Sweden, 2014. Association for Computational Linguistics.

[166] Meishan Zhang, Yue Zhang, and Guohong Fu. Transition-based neural word segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 421–431, Berlin, Germany, 2016. Association for Computational Linguistics.

[167] Qi Zhang, Xiaoyu Liu, and Jinlan Fu. Neural networks incorporating dictionaries for Chinese word segmentation. In *Proceedings of the 32nd AAAI*

Conference on Artificial Intelligence, New Orleans, Louisiana, USA, 2018. AAAI Press.

[168] Yue Zhang and Stephen Clark. 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*, pages 843–852, Cambridge, Massachusetts, USA, 2010. Association for Computational Linguistics.

[169] Yue Zhang and Stephen Clark. Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, 37(1):105–151, 2011.

[170] Hai Zhao and Chunyu Kit. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In *Proceedings of the 6th SIGHAN Workshop on Chinese Language Processing*, pages 106–111, 2008.

[171] Lujun Zhao, Qi Zhang, Peng Wang, and Xiaoyu Liu. 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*, pages 4602–4608, Stockholm, Sweden, 2018. AAAI Press.

[172] Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. Deep learning for Chinese word segmentation and POS tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 647–657, Seattle, Washington, USA, 2013. Association for Computational Linguistics.

[173] Hao Zhou, Zhenting Yu, Yue Zhang, Shujian Huang, Xinyu Dai, and Jiajun Chen. Word-context character embeddings for Chinese word segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 760–766, Copenhagen, Denmark, 2017. Association for Computational Linguistics.

Appendix

A. Calculation Example of Word Summary Vectors

We show a calculation example of WAVG and WCON-based word summary vectors of $x_4 =$“本” in sentence $\boldsymbol{x}$ in Figure 3.1, which is shown again in Figure A.1. Assuming the set of candidate words $\mathcal{W}_x = \{w_1, \ldots, w_8\}$ in Figure A.1 and the maximum word length $K = 4$, $L = \sum_{k=1}^{K} k = 10$ and the list of words $\mathcal{W}'_{x,i=4}$ containing x_4 is as follows:

$$\begin{aligned}
\mathcal{W}'_{x,4} &= \bigcup_{k=1}^{4} \bigcup_{p=-(k-1)}^{0} \{x_{4+p:4+p+k-1}\} \\
&= \bigcup_{p=0}^{0} \{x_{4+p:4+p}\} \cup \bigcup_{p=-1}^{0} \{x_{4+p:4+p+1}\} \cup \bigcup_{p=-2}^{0} \{x_{4+p:4+p+2}\} \cup \bigcup_{p=-3}^{0} \{x_{4+p:4+p+3}\} \\
&= \{x_{4:4}, x_{3:4}, x_{4:5}, x_{2:4}, x_{3:5}, x_{4:6}, x_{1:4}, x_{2:5}, x_{3:6}, x_{4:7}\}.
\end{aligned}$$

$\boldsymbol{x}$	i	1	2	3	4	5					
	x_i	彼	は	日	本	人					
$\mathcal{W}_x$	j	1	2	3	4	5	6	7	8		
	w_j	彼	は	日	本	人	日本	本人	日本人		
$\mathcal{W}'_{x,i=4}$	l	1	2	3	4	5	6	7	8	9	10
	w'_l	$x_{4:4}$ 本	$x_{3:4}$ 日本	$x_{4:5}$ 本人	$x_{2:4}$ は日本	$x_{3:5}$ 日本人	$x_{4:6}$ N/A	$x_{1:4}$ 彼は日本	$x_{2:5}$ は日本人	$x_{3:6}$ N/A	$x_{4:7}$ N/A
	i_l	4	6	7	-	8	-	-	-	-	-

Figure A.1. Illustration of sentence $\boldsymbol{x}$, list of candidate words $\mathcal{W}_x$, and list of words $\mathcal{W}'_{x,i=4}$ ordered by length.

Because $\delta_{ij} = 0$ for $i = 4$ and $j \notin \{4, 6, 7, 8\}$, weight α_{ij} in Eq. (3.2) is calculated as follows:

$$\alpha_{4,j} = \begin{cases} \dfrac{\exp(u_{4,j})}{\sum_{k \in \{4,6,7,8\}} \exp(u_{4,k})} & (j \in \{4, 6, 7, 8\}) \\ 0 & (\text{otherwise}). \end{cases}$$

On the basis on the correspondence between l and i_l as illustrated in Figure A.1, the both types of summary vectors of x_4 in Eqs. (3.3) and (3.4) are calculated as follows:

$$\begin{aligned} \mathrm{WAVG}(x_4, \mathcal{W}_x) &= \sum_{j=1}^{8} \alpha_{4,j} \boldsymbol{e}_j^w = \sum_{j \in \{4,6,7,8\}} \alpha_{4,j} \boldsymbol{e}_j^w \\ &= \alpha_{4,4} \boldsymbol{e}_4^w + \alpha_{4,6} \boldsymbol{e}_6^w + \alpha_{4,7} \boldsymbol{e}_7^w + \alpha_{4,8} \boldsymbol{e}_8^w \ , \\ \mathrm{WCON}(x_4, \mathcal{W}_x) &= \bigoplus_{l=1}^{L} \alpha_{4,i_l} \boldsymbol{e}_{i_l}^w \\ &= [\alpha_{4,4} \boldsymbol{e}_4^w ; \alpha_{4,6} \boldsymbol{e}_6^w ; \alpha_{4,7} \boldsymbol{e}_7^w ; \boldsymbol{0} ; \alpha_{4,8} \boldsymbol{e}_8^w ; \boldsymbol{0} ; \boldsymbol{0} ; \boldsymbol{0} ; \boldsymbol{0} ; \boldsymbol{0}] \ . \end{aligned}$$

B. Model Parameters

In Table B.1, we show details on the numbers of parameters in the baseline and proposed model variants in Chapter 3 when the hyperparameter values in Table 3.2 were used.

	Parameters	BASE	AVG	WAVG	CON	WCON
Char embedding (Avg.)	E_c	963K	963K	963K	963K	963K
Word embedding (Avg.)	E_w	0	15.2M	15.2M	15.2M	15.2M
BiLSTM	$W_i, W_o, W_f, W_t, U_i, U_o, U_f, U_t, \boldsymbol{b}_i, \boldsymbol{b}_o, \boldsymbol{b}_f, \boldsymbol{b}_t$ for each direction and layer	13.0M	13.0M	13.0M	13.0M	13.0M
Bilinear	W_a	0	0	360K	0	3.6M
Affine+CRF	$W_s, \boldsymbol{b}_s, A$	4.8K	6.0K	6.0K	16.8K	16.8K
Total (Avg.)	θ	13.9M	29.1M	29.5M	29.1M	32.7M

Table B.1. The numbers of parameters in model variants. Those in the character and word embedding matrices are average values for the three datasets: BCCWJ, JDC, and JMC.

OC01_{00001,00002,00003,00004,00005,00006,00007}	OY11_{00106,00242}
OC02_{00001,00002,00003,00004,00006,00007,00008}	OY12_00005
OC03_{00001,00005}	OY14_{00047,00236}
OC04_{00001,,00002,00003}	OY15_{00014,00054,00225}
OC05_{00001,00003,00004,00006}	PB11_00006
OC06_{00001,00008}	PB12_00001
OC08_{00001,00002,00004,00006}	PB22_00002
OC09_{00001,00002,00003,00004,00006,00008}	PB43_00001
OC10_{00001,00003,00005,00006,00007}	PB59_00001
OC11_{00001,00002,00004,00005,00006,00007}	PM11_00002
OC12_{00002,00003,00004,00005,00006,00007,00008}	PM24_00003
OC13_{00001,00002,00003,00004,00005,00006,00007,00008}	PN1a_00002
OC14_{00001,00003,00004,00005,00006,00007,00008}	PN1d_{00001,00002}
OC15_{00001,00002,00004,00006,00007,00008}	PN1f_00002
OW6X_{00000,00002,00003,00007,00008,00009,00011,00013}	PN1g_00002
OY01_{00082,00137,00148,00185}	PN2c_00002
OY02_00095	PN2g_00002
OY04_{00001,00027,00173}	PN3b_00001
OY06_{00060,00146,00168}	PN3c_00002
OY07_{00097,00135,00164}	PN4b_00001
OY08_{00115,00137,00156,00180,00186,00189,00198}	PN4c_{00001,00002}
OY09_{00008,00255}	PN4f_00001
OY10_{00050,00062,00067}	

Table C.2. Document IDs in the BCCWJ ClassA-1 set.

C. Document IDs in the BCCWJ Test Set

For the experiments in Chapter 3, we used sentences in the ClassA-1 documents[77] in the BCCWJ core data as the test set. Table C.2 shows the document IDs in the ClassA-1 set.

D. Selection of Closest Standard Form

The process of selecting the closest standard form, which is mentioned in §6.2.1, comprises the following steps. Here, Let w_j be a word and S_j be the set of standard forms of w_j, and we define four character types of a word: “hiragana-only,” “katakana-only,” “kanji-kana-mixed,” and “other.”

1. If S_j contains standard forms with the same character type as w_j, those standard forms are prioritized; standard forms with different character types

[77] `http://www.lsta.media.kyoto-u.ac.jp/resource/data/word-dep/`

are removed from S_j.

2. If S_j contains only standard forms with different character types from w_j, the standard forms with the same character type as the standard form occurring most frequently in a corpus are retained, and others are removed from S_j.

3. The standard form with the most characters that are aligned to w_j is selected as the closest standard form $s_j^\star$. Character alignment between w_j and $s \in S_j$ is calculated, to find the longest matching substrings recursively, until any substrings in w_j and s are not matched.

E. Variant Generation Rules

For variant pair acquisition in Chapter 6, we define ten rules in Table E.3 to generate nonstandard form candidates from standard forms. Rule 2 interchanges お↔を, じ↔ぢ, ず↔づ, ぶ↔ゔ, オ↔ヲ, ジ↔ヂ, ズ↔ヅ, or ブ↔ヴ as characters with the same pronunciation. We generate multiple variants from an original word using any combination of applicable rules in $\{0,1\} \times \{0,2,7\} \times \{0,3,8\} \times \{0,4,5,6\} \times \{0,9,10\}$, where 0 indicates that no rule is applied.

ID	Rule	Sub-rule	Original	Variant
1	Change of character type	(a) Hiragana to katakana	たいへん *taihen* ‘hard’	タイヘン
		(b) Katakana to hiragana	スーパー *sūpā* ‘super’	すーぱー
		(c) Kana-kanji mixed to hiragana	疲労 *hirō* ‘fatigue’	ひろう
		(d) Kana-kanji mixed to katakana	苦手 *nigate* ‘weak’	ニガテ
2	Sub. with same pronunciation character	–	マジ *maji* ‘really’	マヂ
3	Sub. with mora consonant	(a) “です”	です *desu* (copula)	っす
		(b) Adjective ends with “い”	広い *hiroi* ‘wide’	広っ
		(c) Verb ends with “う”	行こう *ikō* ‘go’	行こっ
4	Sub. with uppercase kana	–	ちょっと *chotto* ‘bit’	ちよつと
$5^{S,I}$	Sub. with lowercase kana	–	いや *iya* ‘unpleasant’	ぃゃ
$6^{S,I}$	Sub. of vowel with long sound	–	楽しい *tanoshī* ‘fun’	楽しー
7^{I}	Sub. of vowel sequence	(a) Adjective ends with -*ai* to *ē*	うるさい *uruasi* ‘loud’	うるせえ *urusē*
		(b) Adjective ends with -*ui* to *ī*	わるい *warui* ‘bad’	わりい *warī*
		(c) Adjective ends with -*oi* to *ē*	おそい *osoi* ‘late’	おせえ *osē*
		(d) Word ends with -*o*う to *o*お	そう *sō* ‘so’	そお
		(e) “言う/いう” to ゆう	言い *ī* ‘say’	ゆい *yui*
8^{I}	Deletion of tail vowel	(a) Adjective ends with “い”	ひどい *hidoi* ‘terrible’	ひど *hido*
		(b) Word ends with “う”	だろう *darō* (copula)	だろ *daro*
9^{I}	Insertion of mora consonant	(a) Into the middle	きつい *kitsui* ‘tough’	きっつい *kittsui*
		(b) Into the end	けど *kedo* ‘but’	けどっ
$10^{S,I}$	Insertion of long sound	(a) Long sound sym. into the middle	大きい *ōkī* ‘big’	大きーい *ōkīi*
		(b) Long sound sym. into the end	正解 *seikai* ‘answer’	正解ー *seikaī*
		(d) Uppercase vowel into the middle	かなり *kanari* ‘quite’	かなあり *kanāri*
		(c) Uppercase vowel into the end	強い *tsuyoi* ‘strong’	強いい *tsuyoī*
		(e) Lowercase vowel into the middle	ずっと *zutto* ‘always’	ずぅっと *zūtto*
		(f) Lowercase vowel into the end	ます *masu* (copula)	ますぅ *masū*

Table E.3. Variant generation rules and examples of generated variants. “Sub.” indicates substitution. The IDs with “S” and “I” indicate that similar rules were used in Sasano et al. [112] and Ikeda et al. [47], respectively.

List of Publications

[A] Shohei Higashiyama, Masao Utiyama, Eiichiro Sumita, Masao Ideuchi, Yoshiaki Oida, Yohei Sakamoto, and Isaac Okada. 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*, pages 2699–2709, Minneapolis, Minnesota. Association for Computational Linguistics.

[B] Shohei Higashiyama, Masao Utiyama, Eiichiro Sumita, Masao Ideuchi, Yoshiaki Oida, Yohei Sakamoto, Isaac Okada, and Yuji Matsumoto. 2020. Character-to-Word Attention for Word Segmentation. *Journal of Natural Language Processing*, 27(3):499–530. (C) The Association for Natural Language Processing, (Licensed under CC BY 4.0) `https://creativecommons.org/licenses/by/4.0/` [論文賞]

[C] Shohei Higashiyama, Masao Utiyama, Yuji Matsumoto, Taro Watanabe, and Eiichiro Sumita. 2020. Auxiliary Lexicon Word Prediction for Cross-Domain Word Segmentation. *Journal of Natural Language Processing*, 27(3):573–598. (C) The Association for Natural Language Processing, (Licensed under CC BY 4.0) `https://creativecommons.org/licenses/by/4.0/`

[D] Shohei Higashiyama, Masao Utiyama, Taro Watanabe, and Eiichiro Sumita. 2021. User-generated text corpus for evaluating Japanese morphological analysis and lexical normalization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5532–5541, Online. Association for Computational Linguistics.

[E] Shohei Higashiyama, Masao Utiyama, Taro Watanabe, and Eiichiro Sumita. 2021. A Text Editing Approach to Joint Japanese Word Segmentation, POS Tagging, and Lexical Normalization. In *Proceedings of the 7th Workshop on Noisy User-generated Text*, pages 67–80, Online. Association for Computational Linguistics. [Best Paper Award]